# Table des matières

<table>
<thead>
<tr>
<th>Copyleft</th>
<th>62</th>
</tr>
</thead>
<tbody>
<tr>
<td>tAccessBulkExec</td>
<td>64</td>
</tr>
<tr>
<td>Propriétés du tAccessBulkExec Standard</td>
<td>64</td>
</tr>
<tr>
<td>Scénarios associés</td>
<td>67</td>
</tr>
<tr>
<td>tAccessClose</td>
<td>68</td>
</tr>
<tr>
<td>Propriétés du tAccessClose Standard</td>
<td>68</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>69</td>
</tr>
<tr>
<td>tAccessCommit</td>
<td>70</td>
</tr>
<tr>
<td>Propriétés du tAccessCommit Standard</td>
<td>70</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>71</td>
</tr>
<tr>
<td>tAccessConnection</td>
<td>72</td>
</tr>
<tr>
<td>Propriétés du tAccessConnection Standard</td>
<td>72</td>
</tr>
<tr>
<td>Scénario : Insérer des données dans des tables parent/enfant</td>
<td>73</td>
</tr>
<tr>
<td>tAccessInput</td>
<td>78</td>
</tr>
<tr>
<td>Propriétés du tAccessInput Standard</td>
<td>78</td>
</tr>
<tr>
<td>Scénarios associés</td>
<td>81</td>
</tr>
<tr>
<td>tAccessOutput</td>
<td>82</td>
</tr>
<tr>
<td>Propriétés du tAccessOutput Standard</td>
<td>82</td>
</tr>
<tr>
<td>Scénarios associés</td>
<td>87</td>
</tr>
<tr>
<td>tAccessOutputBulk</td>
<td>88</td>
</tr>
<tr>
<td>Propriétés du tAccessOutputBulk Standard</td>
<td>88</td>
</tr>
<tr>
<td>Scénarios associés</td>
<td>90</td>
</tr>
<tr>
<td>tAccessOutputBulkExec</td>
<td>91</td>
</tr>
<tr>
<td>Propriétés du tAccessOutputBulkExec Standard</td>
<td>91</td>
</tr>
<tr>
<td>Scénarios associés</td>
<td>94</td>
</tr>
<tr>
<td>tAccessRollback</td>
<td>95</td>
</tr>
<tr>
<td>Propriétés du tAccessRollback Standard</td>
<td>95</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>96</td>
</tr>
<tr>
<td>tAccessRow</td>
<td>97</td>
</tr>
<tr>
<td>Propriétés du tAccessRow Standard</td>
<td>97</td>
</tr>
<tr>
<td>Scénarios associés</td>
<td>101</td>
</tr>
<tr>
<td>tAddCRCRow</td>
<td>102</td>
</tr>
<tr>
<td>Propriétés du tAddCRCRow Standard</td>
<td>102</td>
</tr>
<tr>
<td>Scénario : Ajouter une clé de substitution à un fichier</td>
<td>103</td>
</tr>
<tr>
<td>tAddLocationFromIP</td>
<td>106</td>
</tr>
<tr>
<td>Propriétés du tAddLocationFromIP Standard</td>
<td>106</td>
</tr>
<tr>
<td>Scénario : Identifier la localisation géographique d’une adresse IP</td>
<td>107</td>
</tr>
<tr>
<td>tAdvancedFileOutputXML</td>
<td>110</td>
</tr>
<tr>
<td>Propriétés du tAdvancedFileOutputXML Standard</td>
<td>110</td>
</tr>
<tr>
<td>Définir un arbre XML</td>
<td>114</td>
</tr>
<tr>
<td>Mapping de données XML</td>
<td>115</td>
</tr>
<tr>
<td>Définir le statut du nœud</td>
<td>115</td>
</tr>
</tbody>
</table>
Scénario : Créer un fichier XML à l'aide d'une boucle..............................................................116

tAggregateRow................................................................................................................................................................122
Propriétés du tAggregateRow Standard.................................................................................................................122
Agrégé des valeurs et trier des données..................................................................................................................125

tAggregateSortedRow....................................................................................................................................................128
Propriétés du tAggregateSortedRow Standard........................................................................................................128
Scénario : trier et agréger les données d’entrée.................................................................................................130

tAmazonAuroraClose.........................................................................................................................................................136
Propriétés du tAmazonAuroraClose Standard........................................................................................................136
Scénario associé..................................................................................................................................................137

tAmazonAuroraCommit....................................................................................................................................................138
Propriétés du tAmazonAuroraCommit Standard.....................................................................................................138
Scénario associé..................................................................................................................................................140

tAmazonAuroraConnection..............................................................................................................................................141
Propriétés du tAmazonAuroraConnection Standard..............................................................................................141
Scénario associé..................................................................................................................................................143

tAmazonAuroraInput.......................................................................................................................................................144
Propriétés du tAmazonAuroraInput Standard...........................................................................................................144
Scénario : Gestion des données avec Amazon Aurora.........................................................................................148

tAmazonAuroraOutput.....................................................................................................................................................154
Propriétés du tAmazonAuroraOutput Standard.........................................................................................................154
Scénario associé..................................................................................................................................................161

tAmazonAuroraRollback..................................................................................................................................................162
Propriétés du tAmazonAuroraRollback Standard.....................................................................................................162
Scénario associé..................................................................................................................................................163

tAmazonEMRListInstances.........................................................................................................................................164
Propriétés du tAmazonEMRListInstances Standard..............................................................................................164
Scénario associé..................................................................................................................................................166

tAmazonEMRManage.......................................................................................................................................................167
Propriétés du tAmazonEMRManage Standard........................................................................................................167
Gérer un cluster Amazon EMR..................................................................................................................................171

tAmazonEMRResize.........................................................................................................................................................177
Propriétés du tAmazonEMRResize Standard.............................................................................................................177
Scénario associé..................................................................................................................................................179

tAmazonMysqlClose.........................................................................................................................................................180
Propriétés du tAmazonMysqlClose Standard........................................................................................................180
Scénario associé..................................................................................................................................................181

tAmazonMysqlCommit.....................................................................................................................................................182
Propriétés du tAmazonMysqlCommit Standard........................................................................................................182
Scénario associé..................................................................................................................................................184

tAmazonMysqlConnection.............................................................................................................................................185
Propriétés du tAmazonMysqlConnection Standard...............................................................................................185
Scénario associé..................................................................................................................................................187

tAmazonMysqlInput..........................................................................................................................................................188
Propriétés du tAmazonMysqlInput Standard..........................................................................................................188
Scénario associé................................................................................................................................. 245

**tAS400Input**........................................................................................................................................ 246
- Propriétés du tAS400Input Standard.................................................................................................. 246
- Scénario : Gérer des données à l'aide de AS/400.............................................................................. 249
- Scénarios associés............................................................................................................................. 252

**tAS400LastInsertId**.......................................................................................................................... 254
- Propriétés du tAS400LastInsertId Standard....................................................................................... 254
- Scénario associé................................................................................................................................ 255

**tAS400Output**................................................................................................................................. 256
- Propriétés du tAS400Output Standard.................................................................................................. 256
- Scénarios associés............................................................................................................................. 261

**tAS400Rollback**............................................................................................................................... 262
- Propriétés du tAS400Rollback Standard............................................................................................ 262
- Scénario associé................................................................................................................................ 263

**tAS400Row**....................................................................................................................................... 264
- Propriétés du tAS400Row Standard.................................................................................................... 264
- Scénarios associés............................................................................................................................. 267

**tAssert**............................................................................................................................................. 269
- Propriétés du tAssert Standard............................................................................................................ 269
- Scénario 1 : Obtenir le statut des commandes d'un produit (quotidiennement) par rapport à un nombre fixe................................................................................................................................. 270
- Scénario 2 : Paramétrer une condition assertive pour l'exécution d'un Job...................................... 273

**tAssertCatcher**............................................................................................................................... 279
- Propriétés du tAssertCatcher Standard............................................................................................. 279
- Scénario associé................................................................................................................................ 281

**tAzureStorageConnection**.............................................................................................................. 282
- Propriétés du tAzureStorageConnection Standard............................................................................ 282
- Scénario associé................................................................................................................................ 283

**tAzureStorageContainerCreate**..................................................................................................... 284
- Propriétés du tAzureStorageContainerCreate Standard.................................................................... 284
- Scénario : Créer un conteneur dans Azure Storage........................................................................... 286

**tAzureStorageContainerDelete**...................................................................................................... 290
- Propriétés du tAzureStorageContainerDelete Standard.................................................................... 290
- Scénario associé................................................................................................................................ 292

**tAzureStorageContainerExist**....................................................................................................... 293
- Propriétés du tAzureStorageContainerExist Standard....................................................................... 293
- Scénario associé................................................................................................................................ 295

**tAzureStorageContainerList**.......................................................................................................... 296
- Propriétés du tAzureStorageContainerList Standard......................................................................... 296
- Scénario associé................................................................................................................................ 298

**tAzureStorageDelete**....................................................................................................................... 299
- Propriétés du tAzureStorageDelete Standard.................................................................................... 299
- Scénario associé................................................................................................................................ 301

**tAzureStorageGet**......................................................................................................................... 302
- Propriétés du tAzureStorageGet Standard........................................................................................ 302
Scénario : Récupérer des fichiers d’un conteneur Azure Storage ................................................................. 304

tAzureStorageInputTable................................................................................................................................. 312
  Propriétés du tAzureStorageInputTable Standard......................................................................................... 312
  Gérer les données avec Microsoft Azure Table Storage .............................................................................. 315

tAzureStorageList........................................................................................................................................... 322
  Propriétés du tAzureStorageList Standard................................................................................................. 322
  Scénario associé............................................................................................................................................... 325

tAzureStorageOutputTable............................................................................................................................... 326
  Propriétés du tAzureStorageOutputTable Standard..................................................................................... 326
  Scénario associé............................................................................................................................................... 330

tAzureStoragePut........................................................................................................................................... 331
  Propriétés du tAzureStoragePut Standard.................................................................................................... 331
  Scénario associé............................................................................................................................................... 333

tAzureStorageQueueCreate............................................................................................................................ 334
  Propriétés du tAzureStorageQueueCreate Standard.................................................................................... 334
  Scénario associé............................................................................................................................................... 335

tAzureStorageQueueDelete............................................................................................................................ 336
  Propriétés du tAzureStorageQueueDelete Standard...................................................................................... 336
  Scénario associé............................................................................................................................................... 337

tAzureStorageQueueInput............................................................................................................................... 338
  Propriétés du tAzureStorageQueueInput Standard....................................................................................... 338
  Scénario associé............................................................................................................................................... 341

tAzureStorageQueueInputLoop.......................................................................................................................... 342
  Propriétés du tAzureStorageQueueInputLoop Standard.................................................................................. 342
  Scénario associé............................................................................................................................................... 345

tAzureStorageQueueList.................................................................................................................................. 346
  Propriétés du tAzureStorageQueueList Standard.......................................................................................... 346
  Scénario associé............................................................................................................................................... 348

tAzureStorageQueueOutput............................................................................................................................. 349
  Propriétés du tAzureStorageQueueOutput Standard...................................................................................... 349
  Scénario associé............................................................................................................................................... 351

tAzureStorageQueuePurge............................................................................................................................... 352
  Propriétés du tAzureStorageQueuePurge Standard......................................................................................... 352
  Scénario associé............................................................................................................................................... 353

tBarChart....................................................................................................................................................... 354
  Propriétés du tBarChart Standard.................................................................................................................. 354
  Scénario : Créer un diagramme en barres à partir de données d’entrée ........................................................ 356

 tBigQueryBulkExec......................................................................................................................................... 364
  Propriétés du tBigQueryBulkExec Standard.................................................................................................. 364
  Scénario associé............................................................................................................................................... 367

tBigQueryInput................................................................................................................................................ 368
  Propriétés du tBigQueryInput Standard.......................................................................................................... 368
  Scénario : Exécuter une requête dans BigQuery.......................................................................................... 370

tBigQueryOutput............................................................................................................................................ 374
  Propriétés du tBigQueryOutput Standard....................................................................................................... 374
Scénario : Ecrire des données dans BigQuery.......................................................................................................................... 377

**tBigQueryOutputBulk**........................................................................................................................................................................ 383
  Propriétés du tBigQueryOutputBulk Standard......................................................................................................................... 383
  Scénario associé................................................................................................................................................................................. 385

**tBonitaDeploy**.................................................................................................................................................................................. 386
  Propriétés du tBonitaDeploy Standard........................................................................................................................................... 386
  Scénario associé.................................................................................................................................................................................. 388

**tBonitaInstantiateProcess**............................................................................................................................................................... 389
  Propriétés du tBonitaInstantiateProcess Standard.................................................................................................................... 389
  Scénario 1 : Exécuter un processus Bonita via un Job Talend.................................................................................................. 392
  Scénario 2 : Ecrire en sortie l'UUID de l'instance du processus via le lien Row > Main............................................................. 398

**tBoxConnection**................................................................................................................................................................................... 400
  Propriétés du tBoxConnection Standard........................................................................................................................................ 400
  Scénario associé................................................................................................................................................................................... 401

**tBoxCopy**.......................................................................................................................................................................................... 402
  Propriétés du tBoxCopy Standard.................................................................................................................................................. 402
  Scénario associé................................................................................................................................................................................... 404

**tBoxDelete**.......................................................................................................................................................................................... 405
  Propriétés du tBoxDelete Standard.................................................................................................................................................. 405
  Scénario associé................................................................................................................................................................................... 406

**tBoxGet**............................................................................................................................................................................................ 407
  Propriétés du tBoxGet Standard..................................................................................................................................................... 407
  Scénario associé................................................................................................................................................................................... 409

**tBoxList**............................................................................................................................................................................................ 410
  Propriétés du tBoxList Standard..................................................................................................................................................... 410
  Scénario associé................................................................................................................................................................................... 412

**tBoxPut**............................................................................................................................................................................................ 413
  Propriétés du tBoxPut Standard..................................................................................................................................................... 413
  Scénario : Charger et télécharger des fichiers depuis Box........................................................................................................ 415

**tBufferInput**....................................................................................................................................................................................... 418
  Propriétés du tBufferInput Standard.............................................................................................................................................. 418
  Scénario : Récupérer les données mises en mémoire tampon...................................................................................................... 419

**tBufferOutput**....................................................................................................................................................................................... 422
  Propriétés du tBufferOutput Standard........................................................................................................................................... 422
  Scénario 1 : Mettre des données en mémoire tampon................................................................................................................ 423
  Mettre en mémoire tampon des données à utiliser en tant que système source........................................................................ 426
  Scénario 2 : Mettre les données de sortie en mémoire tampon du serveur d'application Web...................................................... 427
  Scénario 3 : Appeler un Job contenant des variables de contexte à partir de votre navigateur Web......................................... 430
  Scénario 4 : Appeler un Job exporté en tant que service Web dans un autre Job........................................................................ 432

**tCassandraBulkExec**.............................................................................................................................................................................. 435
  Propriétés du tCassandraBulkExec Standard............................................................................................................................... 435
  Scénario associé................................................................................................................................................................................... 437

**tCassandraClose**..................................................................................................................................................................................... 438
  Propriétés du tCassandraClose Standard....................................................................................................................................... 438
  Scénario associé................................................................................................................................................................................... 439
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tCreateTable</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchDBOutput</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchDBInput</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchDBConnection</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchbaseOutput</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchbaseInput</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tConvertType</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tContextDump</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tContextLoad</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCosmosDBBulkLoad</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCosmosDBConnection</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCosmosDBInput</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCosmosDBOutput</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCosmosDBRow</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchbaseInput</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchbaseOutput</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchbaseConnection</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchDBCclose</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchDBConnection</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchDBInput</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchDBOutput</td>
<td>Scénario associé.</td>
</tr>
<tr>
<td>tCouchDBRow</td>
<td>Scénario associé.</td>
</tr>
</tbody>
</table>

Propriétés du **tCreateTable** Standard : 

Scénario : Répliquer des données de la base de données source à la base de données cible.

Propriétés du **tCouchDBOutput** Standard : 

Scénario associé.

Propriétés du **tCouchDBInput** Standard : 

Scénario associé.

Propriétés du **tCouchDBConnection** Standard : 

Scénario associé.

Propriétés du **tCouchbaseOutput** Standard : 

Scénario associé.

Propriétés du **tCouchbaseInput** Standard : 

Scénario associé.

Propriétés du **tConvertType** Standard : 

Scénario : Convertir des types de données Java.

Propriétés du **tContextDump** Standard : 

Scénario associé.

Propriétés du **tContextLoad** Standard : 

Scénario associé.

Propriétés du **tCosmosDBBulkLoad** Standard : 

Scénario associé.

Propriétés du **tCosmosDBConnection** Standard : 

Scénario associé.

Propriétés du **tCosmosDBInput** Standard : 

Scénario associé.

Propriétés du **tCosmosDBOutput** Standard : 

Scénario associé.

Propriétés du **tCosmosDBRow** Standard : 

Scénario associé.

Propriétés du **tCouchbaseInput** Standard : 

Scénario associé.

Propriétés du **tCouchbaseOutput** Standard : 

Scénario associé.

Propriétés du **tCouchDBCclose** Standard : 

Scénario associé.

Propriétés du **tCouchDBConnection** Standard : 

Scénario associé.

Propriétés du **tCouchDBInput** Standard : 

Scénario associé.

Propriétés du **tCouchDBOutput** Standard : 

Scénario : Répliquer des données de la base de données source à la base de données cible.

Propriétés du **tCreateTable** Standard.
Scénario : Créer une nouvelle table dans une base de données MySQL.............................................................................. 579

tCreateTemporaryFile................................................................................................................................................................. 581
Propriétés du tCreateTemporaryFile Standard .......................................................................................................................... 581
Scénario : Créer un fichier temporaire et y écrire des données .............................................................................................. 582

tDB2BulkExec.................................................................................................................................................................................. 588
Propriétés du tDB2BulkExec Standard ........................................................................................................................................ 588
Scénario associé ................................................................................................................................................................................ 594

tDB2Close.......................................................................................................................................................................................... 595
Propriétés du tDB2Close Standard ................................................................................................................................................ 595
Scénario associé ................................................................................................................................................................................ 596

tDB2Commit..................................................................................................................................................................................... 597
Propriétés du tDB2Commit Standard ........................................................................................................................................... 597
Scénario associé ................................................................................................................................................................................ 598

tDB2Connection.............................................................................................................................................................................. 599
Propriétés du tDB2Connection Standard ....................................................................................................................................... 599
Scénarios associés ............................................................................................................................................................................ 601

tDB2Input....................................................................................................................................................................................... 602
Propriétés du tDB2Input Standard ................................................................................................................................................ 602
Scénarios associés ............................................................................................................................................................................ 606

tDB2Output...................................................................................................................................................................................... 607
Propriétés du tDB2Output Standard ............................................................................................................................................ 607
Scénarios associés ............................................................................................................................................................................ 613

tDB2Rollback.................................................................................................................................................................................. 614
Propriétés du tDB2Rollback Standard ........................................................................................................................................ 614
Scénario associé ................................................................................................................................................................................ 615

tDB2Row......................................................................................................................................................................................... 616
Propriétés du tDB2Row Standard ................................................................................................................................................ 616
Scénarios associés ............................................................................................................................................................................ 620

tDB2SCD......................................................................................................................................................................................... 621
Propriétés du tDB2SCD Standard ................................................................................................................................................ 621
Scénario associé ................................................................................................................................................................................ 624

tDB2SCDELT................................................................................................................................................................................... 625
Propriétés du tDB2SCDELT Standard .......................................................................................................................................... 625
Scénario associé ................................................................................................................................................................................ 629

tDB2SP......................................................................................................................................................................................... 630
Propriétés du tDB2SP Standard ................................................................................................................................................... 630
Scénario associé ................................................................................................................................................................................ 633

Composants de bases de données dynamiques .......................................................................................................................... 634

tDBBulkExec.................................................................................................................................................................................... 635
Propriétés du tDBBulkExec Standard ......................................................................................................................................... 635

tDBClose......................................................................................................................................................................................... 637
Propriétés du tDBClose Standard ................................................................................................................................................ 637

tDBColumnList.............................................................................................................................................................................. 639
Propriétés du tDBColumnList Standard ....................................................................................................................................... 639

tDBCommit.................................................................................................................................................................................... 640
Scénario : Utiliser .NET dans Talend ................................................................................................................................ 680

Propriétés du tDotNETRow Standard ................................................................................................................................ 678

Scénario associé ........................................................................................................................................................................ 677

Propriétés du tDotNETInstantiate Standard ...................................................................................................................676

Scénarios associés .................................................................................................................................................................... 675

Propriétés du tDie Standard .................................................................................................................................................674

Scénarios associés .................................................................................................................................................................... 674

Propriétés du tDenormalize SortedRow Standard............................................................................................................. 669

Scénario : Regrouper des lignes triées .................................................................................................................................. 670

tDenormalize........................................................................................................................................................................ 663

Propriétés du tDenormalize Standard .................................................................................................................................. 663

Scénario : Dénormaliser une colonne .......................................................................................................................................... 664

Scénario 2 : Dénormaliser plusieurs colonnes ................................................................................................................................. 666

tDenormalizeSortedRow........................................................................................................................................................... 669

Propriétés du tDenormalizeSortedRow Standard ............................................................................................................. 669

Scénario : Regrouper des lignes triées .................................................................................................................................. 670

tDie......................................................................................................................................................................................... 674

Propriétés du tDie Standard .................................................................................................................................................674

Scénarios associés .................................................................................................................................................................... 675

tDotNETInstantiate ................................................................................................................................................................. 676

Propriétés du tDotNETInstantiate Standard ............................................................................................................................. 676

Scénario associé ......................................................................................................................................................................... 677

tDotNETRow............................................................................................................................................................................. 678

Propriétés du tDotNETRow Standard .......................................................................................................................................... 678

Scénario : Utiliser .NET dans Talend .......................................................................................................................................... 680

tDropboxConnection.................................................................................................................................................................. 685
Propriétés du tDropboxConnection Standard.................................................................685
Scénario associé..............................................................................................................686

tDropboxDelete............................................................................................................687
Propriétés du tDropboxDelete Standard........................................................................687
Scénario associé...............................................................................................................688

tDropboxGet...................................................................................................................689
Propriétés du tDropboxGet Standard..............................................................................689
Scénario associé...............................................................................................................690

tDropboxList....................................................................................................................691
Propriétés du tDropboxList Standard..............................................................................691
Scénario associé...............................................................................................................692

tDropboxPut.....................................................................................................................693
Propriétés du tDropboxPut Standard...............................................................................693
Scénario : Télécharger des fichiers dans Dropbox.........................................................695

tDTDValidator................................................................................................................701
Propriétés du tDTDValidator Standard............................................................................701
Scénario : Valider un fichier XML..................................................................................702

tDynamoDBInput.............................................................................................................705
Propriétés du tDynamoDBInput Standard.......................................................................705
Scénario associé...............................................................................................................709

tDynamoDBOutput..........................................................................................................710
Propriétés du tDynamoDBOutput Standard......................................................................710
Scénario associé...............................................................................................................713

tEDIFACTtoXML..............................................................................................................714
Propriétés du tEDIFACTtoXML Standard.........................................................................714
Scénario : Passer d’EDIFACT à XML.............................................................................715

tELTGreenplumInput.......................................................................................................718
Propriétés du tELTGreenplumInput Standard.................................................................718
Scénarios associés..........................................................................................................719

tELTGreenplumMap.........................................................................................................721
Propriétés du tELTGreenplumMap Standard.................................................................721
Scénario : Mapper les données à l’aide d’une jointure implicite simple............................724
Scénario associé...............................................................................................................730

tELTGreenplumOutput..................................................................................................731
Propriétés du tELTGreenplumOutput Standard.............................................................731
Scénarios associés..........................................................................................................733

tELTHiveInput................................................................................................................734
Propriétés du tELTHiveInput Standard..........................................................................734
Scénarios associés..........................................................................................................736

tELTHiveMap..................................................................................................................737
Propriétés du tELTHiveMap Standard............................................................................737
Scénario : Effectuer une jointure sur les colonnes d’une table et les écrire dans Hive................749
Scénarios associés..........................................................................................................757

tELTHiveOutput............................................................................................................758
Propriétés du tELTHiveOutput Standard.......................................................................758
Scénarios associés.......................................................................................................................................................... 760

**tELTJDBCInput**.......................................................................................................................................................... 761
  Propriétés du tELTJDBCInput Standard...................................................................................................................... 761
  Scénarios associés.......................................................................................................................................................... 762

**tELTJDBCMap**.............................................................................................................................................................. 764
  Propriétés du tELTJDBCMap Standard........................................................................................................................ 764
  Agréger des données Snowflake à l'aide de variables de contexte comme noms de tables et de connexion.............................................. 767
  Scénarios associés.......................................................................................................................................................... 771

**tELTJDBCOoutput**......................................................................................................................................................... 772
  Propriétés du tELTJDBCOoutput Standard................................................................................................................... 772
  Scénarios associés.......................................................................................................................................................... 774

**tELTMSSqlInput**............................................................................................................................................................ 776
  Propriétés du tELTMSSqlInput Standard.................................................................................................................... 776
  Scénarios associés.......................................................................................................................................................... 777

**tELTMSSqlMap**............................................................................................................................................................. 778
  Propriétés du tELTMSSqlMap Standard........................................................................................................................ 778
  Scénarios associés.......................................................................................................................................................... 780

**tELTMSSqlOutput**......................................................................................................................................................... 782
  Propriétés du tELTMSSqlOutput Standard.................................................................................................................... 782
  Scénarios associés.......................................................................................................................................................... 784

**tELTMysqlInput**............................................................................................................................................................ 785
  Propriétés du tELTMysqlInput Standard.................................................................................................................... 785
  Scénarios associés.......................................................................................................................................................... 786

**tELTMysqlMap**............................................................................................................................................................. 787
  Propriétés du tELTMysqlMap Standard........................................................................................................................ 787
  Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre.................................................................................. 790
  Scénario 2 : ELT utilisant une table Alias.................................................................................................................. 794
  Scénarios associés.......................................................................................................................................................... 798

**tELTMysqlOutput**.......................................................................................................................................................... 799
  Propriétés du tELTMysqlOutput Standard................................................................................................................... 799
  Scénarios associés.......................................................................................................................................................... 801

**tELTNetezzaInput**.......................................................................................................................................................... 802
  Propriétés du tELTNetezzaInput Standard.................................................................................................................... 802
  Scénarios associés.......................................................................................................................................................... 803

**tELTNetezzaMap**............................................................................................................................................................ 804
  Propriétés du tELTNetezzaMap Standard.................................................................................................................... 804
  Scénarios associés.......................................................................................................................................................... 806

**tELTNetezzaOutput**......................................................................................................................................................... 808
  Propriétés du tELTNetezzaOutput Standard................................................................................................................... 808
  Scénarios associés.......................................................................................................................................................... 808

**tELTOracleInput**............................................................................................................................................................ 811
  Propriétés du tELTOracleInput Standard.................................................................................................................... 811
  Scénarios associés.......................................................................................................................................................... 812

**tELTOracleMap**............................................................................................................................................................. 813
Propriétés du tELTOracleMap Standard................................................................. 813
Scénario : Mettre à jour les lignes d'une base Oracle........................................ 816
Scénario associé.................................................................................. 818
tELTOracleOutput.............................................................................................. 819
Propriétés du tELTOracleOutput Standard..................................................... 819
Scénarios associés..................................................................................... 822
tELTPostgresqlInput......................................................................................... 827
Propriétés du tELTPostgresqlInput Standard................................................ 827
Scénarios associés..................................................................................... 828
tELTPostgresqlMap.......................................................................................... 829
Propriétés du tELTPostgresqlMap Standard................................................ 829
Scénarios associés..................................................................................... 831
tELTPostgresqlOutput...................................................................................... 832
Propriétés du tELTPostgresqlOutput Standard............................................. 832
Scénarios associés..................................................................................... 834
tELTSybaseInput............................................................................................. 835
Propriétés du tELTSybaseInput Standard...................................................... 835
Scénarios associés..................................................................................... 836
tELTSybaseMap.............................................................................................. 837
Propriétés du tELTSybaseMap Standard....................................................... 837
Scénarios associés..................................................................................... 839
tELTSybaseOutput.......................................................................................... 840
Propriétés du tELTSybaseOutput Standard................................................ 840
Scénarios associés..................................................................................... 842
tELTTeradataInput.......................................................................................... 843
Propriétés du tELTTeradataInput Standard................................................ 843
Scénarios associés..................................................................................... 844
tELTTeradataMap............................................................................................ 845
Propriétés du tELTTeradataMap Standard................................................... 845
Scénario : Mapper des données à l'aide d'une sous-requête......................... 848
Scénarios associés................................................................................... 855
tELTTeradataOutput....................................................................................... 856
Propriétés du tELTTeradataOutput Standard.............................................. 856
Scénarios associés..................................................................................... 858
tELTVerticalInput........................................................................................... 859
Propriétés du tELTVerticalInput Standard.................................................. 859
Scénarios associés..................................................................................... 860
tELTVerticalMap............................................................................................ 861
Propriétés du tELTVerticalMap Standard.................................................... 861
Scénarios associés..................................................................................... 863
tELTVerticalOutput........................................................................................ 864
Propriétés du tELTVerticalOutput Standard................................................. 864
Scénarios associés..................................................................................... 866
tESBConsumer............................................................................................... 867
Propriétés du tESBConsumer Standard................................................................................................................. 867
Scénario 1 : Utiliser le composant tESBConsumer pour récupérer une adresse e-mail valide......................... 873
Scénario 2 : Utiliser le tESBConsumer avec des en-têtes SOAP personnalisés....................................................... 881
tESBProviderFault.................................................................................................................................................. 892
Propriétés du tESBProviderFault Standard.............................................................................................................. 892
Scénario : Effectuer une requête sur les nom d'aéroports à partir des codes pays............................................ 893
tESBProviderRequest........................................................................................................................................... 905
Propriétés du tESBProviderRequest Standard......................................................................................................... 905
Scénario : Envoi d'un message via un service sans attente de réponse............................................................... 908
tESBProviderResponse........................................................................................................................................... 918
Propriétés du tESBProviderResponse Standard...................................................................................................... 918
Scénario : Retourner une réponse "Hello world".................................................................................................. 919
tEXABulkExec....................................................................................................................................................... 930
Propriétés du tEXABulkExec Standard....................................................................................................................... 930
Paramètres des différences sources de données importées.................................................................................. 935
Scénario : Importer des données dans une table de base de données EXASolution à partir d'un fichier local CSV......................................................................................................................... 939
tEXAClose.............................................................................................................................................................. 945
Propriétés du tEXAClose Standard.............................................................................................................................. 945
Scénario associé.......................................................................................................................................................... 946
tEXACommit......................................................................................................................................................... 947
Propriétés du tEXACommit Standard............................................................................................................................ 947
Scénario associé.......................................................................................................................................................... 949
tEXAConnection.................................................................................................................................................. 950
Propriétés du tEXAConnection Standard.................................................................................................................. 950
Scénario associé.......................................................................................................................................................... 952
tEXAInput.............................................................................................................................................................. 953
Propriétés du tEXAInput Standard............................................................................................................................... 953
Scénario associé.......................................................................................................................................................... 957
tEXAOutput......................................................................................................................................................... 958
Propriétés du tEXAOutput Standard............................................................................................................................. 958
Scénario associé.......................................................................................................................................................... 964
tEXARollback...................................................................................................................................................... 965
Propriétés du tEXARollback Standard........................................................................................................................... 965
Scénario associé.......................................................................................................................................................... 966
tEXARow.............................................................................................................................................................. 967
Propriétés du tEXARow Standard................................................................................................................................. 967
Scénario associé.......................................................................................................................................................... 970
tEXistConnection................................................................................................................................................. 971
Propriétés du tEXistConnection Standard.................................................................................................................. 971
Scénario associé.......................................................................................................................................................... 972
tEXistDelete...................................................................................................................................................... 973
Propriétés du tEXistDelete Standard............................................................................................................................. 973
Scénario associé.......................................................................................................................................................... 975
tEXistGet......................................................................................................................................................... 976
Scénario : Récupérer des ressources à partir d’un serveur distant de base de données eXist......978

Propriétés du tEXistGet Standard....................................................................................................................976

tEXistList..............................................................................................................................................................980
Propriétés du tEXistList Standard..........................................................................................................................980
Scénario associé.......................................................................................................................................................982

Propriétés du tEXistPut Standard.............................................................................................................................983
Scénario associé.......................................................................................................................................................983

Scénario associé..........................................................................................................................................................984

Scénario associé..........................................................................................................................................................987

Scénario associé..........................................................................................................................................................988

Propriétés du tEXistXUpdate Standard.....................................................................................................................987

Scénario associé..........................................................................................................................................................989

Propriétés du tEXistXQuery Standard.......................................................................................................................985

Scénario associé..........................................................................................................................................................986

Scénario associé..........................................................................................................................................................987

Scénario associé..........................................................................................................................................................988

Scénario associé..........................................................................................................................................................989

Scénario associé..........................................................................................................................................................990

Scénario associé..........................................................................................................................................................992

Scénario associé..........................................................................................................................................................993

Scénario associé..........................................................................................................................................................995

Propriétés du tEXistDelimitedFields Standard.........................................................................................................993

Scénario : Extraire une colonne String délimitée d’une table d’une base de données..........................................995

Propriétés du tEXistJSONFields Standard................................................................................................................1001

Scénario : Récupérer les messages d’erreur lors de l’extraction de données de champs JSON.................................1003
Scénario 2 : Collecter des données de votre réseau social favori...........................................................................1009

Propriétés du tEXistPositionalFields Standard.........................................................................................................1014

Scénario associé..........................................................................................................................................................1016

Propriétés du tEXistRegexpFields Standard............................................................................................................1017

Scénario : Extraire des noms, des domaines et domaines de premier niveau à partir d’adresses e-mail.................1019

Propriétés du tEXistXMLField Standard...................................................................................................................1022

Scénario 1 : Extraire les données XML d’un champ d’une table de base de données.................................................1024
Scénario 2 : Extraire les données valides et les données erronées à partir d’un champ XML dans fichier délimité..........................................................................................................................................................1026

Propriétés du tFileArchive Standard........................................................................................................................1030

Scénario : Zipper des fichiers à l’aide d’un tFileArchive...........................................................................................1033

Propriétés du tFileCompare Standard........................................................................................................................1035

Scénario : Comparer des fichiers dézippés................................................................................................................1036

Propriétés du tFileCopy Standard................................................................................................................................1039

Scénario : Récupérer un fichier de la corbeille..........................................................................................................1041

tFileDelete...............................................................................................................................................................1043
Scénario : Lecture d'un fichier XML multi-structuré ................................................................................................ 1123
Propriétés du tFileInputMSXML Standard .................................................................................................................... 1121
Scénario : Lire des données d'un fichier positionnel ...............................................................................................1117
Propriétés du tFileInputMSPositional Standard ......................................................................................................... 1115
Scénario : Extraire des champs clés d'un e-mail .......................................................................................................1104
Propriétés du tFileInputLDIF Standard ..........................................................................................................................1099
Scénario 4 : Extraire des données JSON d'une URL ................................................................................................. 1095
Scénario 3 : Extraire des données JSON d'un fichier en utilisant XPath ...........................................................1093
Scénario 2 : Extraire des données JSON d'un fichier en utilisant JSONPath ....................................................1091
Scénario 1 : Extraire des données JSON d'un fichier en utilisant le JSONPath sans configurer de nœud de boucle.1088
Scénario : Lire des lignes complètes dans un fichier délimité .............................................................................1082
Propriétés du tFileInputJSON Standard ......................................................................................................................... 1085
Scénario : Afficher le contenu d'un fichier ARFF..................................................................................................1064
Propriétés du tFileInputARFF Standard ......................................................................................................................... 1062
Propriétés du tFileInputDelimited Standard ................................................................................................................1067
Scénario 2 : Lire les données d'un fichier distant en mode stream ........................................................................1072
Scénario : Lire des données d'un fichier positionnel ..............................................................................................1081
Propriétés du tFileInputFullRow Standard ...................................................................................................................1081
Scénario associé ..........................................................................................................................................................1080
Scénario : Afficher le contenu d'un fichier délimité .............................................................................................1076
Propriétés du tFileInputExcel Standard ......................................................................................................................... 1076
Propriétés du tFileInputDelimited Standard ................................................................................................................1067
Scénario 1 : Afficher le contenu d'un fichier délimité.............................................................................................1070
Scénario 1 : Récupérer des données à partir d'un protocole HTTP ........................................................................1055
Scénario 2 : Réutiliser un cookie stocké pour récupérer des fichiers via un protocole HTTP ....................1058
Scénario associé ..........................................................................................................................................................1052
Scénario : Supprimer des fichiers ....................................................................................................................................1044
Propriétés du tFileExist Standard ....................................................................................................................................1046
Scénario : Lire des données d'un fichier positionnel ..............................................................................................1115
Propriétés du tFileInputMSPositional Standard ......................................................................................................... 1115
Scénario : Lire des données d'un fichier délimité ..............................................................................................1067
Propriétés du tFileInputDelimited Standard ................................................................................................................1067
Scénario 1 : Afficher le contenu d'un fichier délimité.............................................................................................1070
Propriétés du tFileInputExcel Standard ......................................................................................................................... 1076
tFileInputPositional ....................................................................................................................... 1126
  Propriétés du tFileInputPositional Standard .................................................................................. 1126
  Transformer un fichier positionnel en fichier XML ................................................................. 1129

tFileInputProperties ..................................................................................................................... 1134
  Propriétés du tFileInputProperties Standard .................................................................................. 1134
  Scénario : Lire et mapper la clé et les valeurs de fichiers properties et alimenter un glossaire ................................................................................................................................................................................. 1135

tFileInputRaw ................................................................................................................................ 1139
  Propriétés du tFileInputRaw Standard ............................................................................................ 1139
  Scénario associé ................................................................................................................................ 1141

tFileInputRegex ................................................................................................................................ 1142
  Propriétés du tFileInputRegex Standard ........................................................................................... 1142
  Scénario : Transformer en fichier Regex en Positional .............................................................. 1144

tFileInputXML ................................................................................................................................ 1147
  Propriétés du tFileInputXML Standard............................................................................................ 1147
  Scénario 1 : Extraire des adresses XML ...................................................................................... 1150
  Scénario 2 : Extraire les données XML erronées dans un flux de rejet ........................................ 1151

tFileList ......................................................................................................................................... 1155
  Propriétés du tFileList Standard .................................................................................................... 1155
  Scénario 1 : Itération sur un répertoire .......................................................................................... 1157
  Scénario 2 : Trouver des fichiers dupliqués entre deux dossiers ................................................... 1160

tFileOutputARFF ............................................................................................................................ 1165
  Propriétés du tFileOutputARFF Standard .......................................................................................... 1165
  Scénario associé ................................................................................................................................ 1168

tFileOutputDelimited ..................................................................................................................... 1169
  Propriétés du tFileOutputDelimited Standard .................................................................................. 1169
  Scénario 1 : Ecrire des données dans un fichier délimité ............................................................ 1172
  Scénario 2 : Utiliser un flux de sortie pour sauvegarder des données dans un fichier local .......... 1177

tFileOutputExcel ............................................................................................................................ 1179
  Propriétés du tFileOutputExcel Standard .......................................................................................... 1179
  Scénario associé ................................................................................................................................ 1182

tFileOutputJSON ............................................................................................................................ 1183
  Propriétés du tFileOutputJSON Standard .......................................................................................... 1183
  Scénario : Ecrire un fichier JSON structuré ................................................................................ 1185

tFileOutputLDIF .............................................................................................................................. 1188
  Propriétés du tFileOutputLDIF Standard .......................................................................................... 1188
  Scénario : Écrire des données d'une table d'une base de données dans un fichier LDIF ............... 1191

tFileOutputMSDelimited ............................................................................................................... 1196
  Propriétés du tFileOutputMSDelimited Standard .......................................................................... 1196
  Scénario associé .............................................................................................................................. 1198

tFileOutputMSPositional ............................................................................................................... 1199
  Propriétés du tFileOutputMSPositional Standard .......................................................................... 1199
  Scénario associé .............................................................................................................................. 1201

tFileOutputMSXML ....................................................................................................................... 1202
  Propriétés du tFileOutputMSXML Standard .................................................................................. 1202
  Définir un arbre XML Multischéma .............................................................................................. 1203
Mapping de données XML à partir de sources multischéma............................................................................... 1205
Définir le statut du nœud................................................................................................................................. 1206
Scénario associé............................................................................................................................................... 1207

**tFirebirdInput**........................................................................................................................................ 1250
Propriétés du tFirebirdInput Standard........................................................................................................... 1250
Scénario associé............................................................................................................................................... 1251

**tFirebirdConnection** .................................................................................................................................. 1247
Propriétés du tFirebirdConnection Standard.................................................................................................. 1247
Scénario associé............................................................................................................................................... 1248

**tFirebirdClose**.......................................................................................................................................... 1243
Propriétés du tFirebirdClose Standard............................................................................................................ 1243
Scénario associé............................................................................................................................................... 1244

**tFilterRow**................................................................................................................................................ 1236
Propriétés du tFilterRow Standard.................................................................................................................. 1236
Scénario associé............................................................................................................................................... 1237

**tFilterColumns**........................................................................................................................................ 1234
Propriétés du tFilterColumns Standard........................................................................................................ 1234
Scénario associé............................................................................................................................................... 1235

**tFileUnarchive**....................................................................................................................................... 1231
Propriétés du tFileUnarchive Standard......................................................................................................... 1231
Scénario associé............................................................................................................................................... 1232

**tFileRowCount**........................................................................................................................................ 1223
Propriétés du tFileRowCount Standard........................................................................................................... 1223
Scénario : Ecrire un fichier dans MySQL si le nombre d'enregistrements correspond à une valeur de référence......................................................................................................................... 1224

**tFileProperties**...................................................................................................................................... 1220
Propriétés du tFileProperties Standard......................................................................................................... 1220
Scénario : Afficher les propriétés d’un fichier traité...................................................................................... 1221

**tFileOutputXML**.................................................................................................................................... 1216
Propriétés du tFileOutputXML Standard........................................................................................................ 1216
Scénarios associés............................................................................................................................................ 1217

**tFileOutputRaw**...................................................................................................................................... 1214
Propriétés du tFileOutputRaw Standard......................................................................................................... 1214
Scénarios associés............................................................................................................................................ 1215

**tFileOutputProperties**.......................................................................................................................... 1212
Propriétés du tFileOutputProperties Standard............................................................................................... 1212
Scénario associé............................................................................................................................................... 1213

**tFileOutputPositional**............................................................................................................................ 1208
Propriétés du tFileOutputPositional Standard................................................................................................. 1208
Scénario associé............................................................................................................................................... 1209

**tFileOutputXML**.................................................................................................................................... 1216
Propriétés du tFileOutputXML Standard........................................................................................................ 1216
Scénarios associés............................................................................................................................................ 1217

**tFileProperties**...................................................................................................................................... 1220
Propriétés du tFileProperties Standard......................................................................................................... 1220
Scénario : Afficher les propriétés d’un fichier traité...................................................................................... 1221

**tFileRowCount**....................................................................................................................................... 1223
Scénario associé............................................................................................................................................... 1224

**tFileTouch**................................................................................................................................................ 1229
Propriétés du tFileTouch Standard................................................................................................................ 1229
Scénario associé............................................................................................................................................... 1230

**tFileUnarchive**....................................................................................................................................... 1231
Propriétés du tFileUnarchive Standard......................................................................................................... 1231
Scénario associé............................................................................................................................................... 1232

**tFilterColumns**...................................................................................................................................... 1234
Propriétés du tFilterColumns Standard........................................................................................................ 1234
Scénario associé............................................................................................................................................... 1235

**tFilterRow**.............................................................................................................................................. 1236
Scénario 1 : Filtrer une liste de noms à l’aide de conditions simples.............................................................. 1238
Scénario 2 : Filtrer une liste de noms via différentes opérations logiques...................................................... 1241

**tFirebirdClose**....................................................................................................................................... 1243
Propriétés du tFirebirdClose Standard........................................................................................................... 1243
Scénario associé............................................................................................................................................... 1244

**tFirebirdCommit**.................................................................................................................................... 1245
Propriétés du tFirebirdCommit Standard....................................................................................................... 1245
Scénario associé............................................................................................................................................... 1246

**tFirebirdConnection**............................................................................................................................... 1247
Propriétés du tFirebirdConnection Standard................................................................................................. 1247
Scénario associé............................................................................................................................................... 1248

**tFirebirdInput**....................................................................................................................................... 1250
Propriétés du tFirebirdInput Standard........................................................................................................... 1250
Scénarios associés..................................................................................................................................................................1391
Scénario : Décrypter un fichier crypté GnuPG et afficher son contenu ............................................................. 1384
Propriétés du tGPGDecrypt Standard..............................................................................................................................1383
Scénario associé......................................................................................................................................................................1384
Propriétés du tFTPTruncate Standard............................................................................................................................1326
Placer des fichiers sur un serveur FTP...............................................................................................................................1316

tGPGDecrypt.......................................................................................................................................................... 1383

Renommer un fichier situé sur un serveur FTP ......................................................................................................... 1323

tFTPTruncate.........................................................................................................................................................1326

Scénario associé......................................................................................................................................................................1326
tFuzzyMatch..............................................................................................................................................................1330

Propriétés du tFuzzyMatch Standard..............................................................................................................................1330
Scénario 1 : Distance de Levenshtein de 0 pour les prénoms..................................................................................1332
Scénario 2 : Distance de Levenshtein de 1 ou 2 pour les prénoms...........................................................................1334
Scénario 3 : Distance métaphonique pour les prénoms.........................................................................................1335

Propriétés du tGoogleDataprocManage Standard......................................................................................................1337

Scénario associé......................................................................................................................................................................1337
Scénario 3 : Distance métaphonique pour les prénoms ......................................................................................... 1335

Propriétés du tGoogleDriveConnection Standard......................................................................................................1340
Méthodes OAuth pour accéder à Google Drive.............................................................................................................1342
Scénario associé......................................................................................................................................................................1352

Propriétés du tGoogleDriveCopy Standard................................................................................................................1353
Scénario associé......................................................................................................................................................................1353
Scénario associé......................................................................................................................................................................1353

Propriétés du tGoogleDriveCreate Standard...........................................................................................................1357
Scénario associé......................................................................................................................................................................1357
Scénario associé......................................................................................................................................................................1357

Propriétés du tGoogleDriveDelete Standard.............................................................................................................1360
Scénario associé......................................................................................................................................................................1360
Scénario associé......................................................................................................................................................................1360

Propriétés du tGoogleDriveGet Standard................................................................................................................1363
Scénario associé......................................................................................................................................................................1363
Scénario associé......................................................................................................................................................................1363

Propriétés du tGoogleDriveList Standard................................................................................................................1367
Scénario associé......................................................................................................................................................................1367
Scénario associé......................................................................................................................................................................1367

Propriétés du tGoogleDrivePut Standard..................................................................................................................1371
Gestion des fichiers avec Google Drive.........................................................................................................................1374

Propriétés du tGoogleDriveList Standard................................................................................................................1367

Scénario associé......................................................................................................................................................................1367
Scénario associé......................................................................................................................................................................1367

Propriétés du tGoogleDriveCreate Standard...........................................................................................................1357

Scénario associé......................................................................................................................................................................1357
Scénario associé......................................................................................................................................................................1357

Propriétés du tGoogleDriveDelete Standard.............................................................................................................1360

Scénario associé......................................................................................................................................................................1360
Scénario associé......................................................................................................................................................................1360

Propriétés du tGoogleDriveGet Standard................................................................................................................1363

Scénario associé......................................................................................................................................................................1363
Scénario associé......................................................................................................................................................................1363

Propriétés du tGoogleDriveList Standard................................................................................................................1367

Scénario associé......................................................................................................................................................................1367
Scénario associé......................................................................................................................................................................1367

Propriétés du tGoogleDrivePut Standard..................................................................................................................1371

Gestion des fichiers avec Google Drive.........................................................................................................................1374

Propriétés du tGPGDecrypt Standard..........................................................................................................................1383

Scénario : Décrypter un fichier crypté GnuPG et afficher son contenu.................................................................1384

Propriétés du tGreenplumBulkExec Standard...........................................................................................................1388
Scénarios associés......................................................................................................................................................................1388

Propriétés du tGreenplumBulkExec Standard...........................................................................................................1388
Scénarios associés......................................................................................................................................................................1388
tGreenplumClose.......................................................................................................................................................... 1393
Propriétés du tGreenplumClose Standard................................................................................................................ 1393
Scénario associé............................................................................................................................................................ 1394

tGreenplumCommit...................................................................................................................................................... 1395
Propriétés du tGreenplumCommit Standard................................................................................................................ 1395
Scénario associé............................................................................................................................................................ 1396

tGreenplumClose.......................................................................................................................................................... 1397
Scénario associé............................................................................................................................................................ 1397
Scénarios associés........................................................................................................................................................ 1398

tGreenplumGPLoad..................................................................................................................................................... 1400
Propriétés du tGreenplumGPLoad Standard................................................................................................................ 1400
Scénario associé............................................................................................................................................................ 1406

tGreenplumInput.......................................................................................................................................................... 1407
Propriétés du tGreenplumInput Standard................................................................................................................... 1407
Scénarios associés........................................................................................................................................................ 1410

tGreenplumOutput...................................................................................................................................................... 1411
Propriétés du tGreenplumOutput Standard.................................................................................................................. 1411
Scénarios associés........................................................................................................................................................ 1416

tGreenplumOutputBulk.............................................................................................................................................. 1417
Propriétés du tGreenplumOutputBulk Standard........................................................................................................ 1417
Scénarios associés......................................................................................................................................................... 1419

tGreenplumOutputBulkExec......................................................................................................................................... 1420
Propriétés du tGreenplumOutputBulkExec Standard.................................................................................................. 1420
Scénarios associés........................................................................................................................................................ 1423

tGreenplumRollback.................................................................................................................................................... 1424
Propriétés du tGreenplumRollback Standard.............................................................................................................. 1424
Scénario associé............................................................................................................................................................ 1425

tGreenplumRow........................................................................................................................................................... 1426
Propriétés du tGreenplumRow Standard...................................................................................................................... 1426
Scénarios associés........................................................................................................................................................ 1430

tGreenplumSCD......................................................................................................................................................... 1431
Propriétés du tGreenplumSCD Standard..................................................................................................................... 1431
Scénario associé............................................................................................................................................................ 1434

tGroovy........................................................................................................................................................................ 1435
Propriétés du tGroovy Standard................................................................................................................................. 1435
Scénarios associés......................................................................................................................................................... 1436

tGroovyFile................................................................................................................................................................. 1437
Propriétés du tGroovyFile Standard............................................................................................................................ 1437
Scénario : Appeler du code Groovy contenu dans un fichier....................................................................................... 1438

tGSBucketCreate....................................................................................................................................................... 1440
Propriétés du tGSBucketCreate Standard................................................................................................................ 1440
Scénario associé............................................................................................................................................................ 1442

tGSBucketDelete......................................................................................................................................................... 1443
Propriétés du tGSBucketDelete Standard.................................................................................................................. 1443
Scénario associé............................................................................................................................................................ 1444
Scénario : Echanger des données clients avec HBase .............................................................................................1501

Propriétés du tHBaseInput Standard .............................................................................................................................. 1494

Filtres HBase............................................................................................................................................................................ 1493

Scénario associé......................................................................................................................................................................1492

Propriétés du tHBaseConnection Standard .................................................................................................................. 1487

Scénario associé......................................................................................................................................................................1486

Propriétés du tHashOutput Standard .............................................................................................................................1482

Scénario : Gérer des fichiers avec Google Cloud Storage ...................................................................................... 1465

Propriétés du tGSPut Standard .........................................................................................................................................1464

Scénario associé......................................................................................................................................................................1463

Propriétés du tGSList Standard .........................................................................................................................................1462

Scénario associé......................................................................................................................................................................1461

Propriétés du tGSDelete Standard ...................................................................................................................................1456

Propriétés du tGSCopy Standard ......................................................................................................................................1453

Propriétés du tGSConnection Standard ......................................................................................................................... 1451

Propriétés du tGSClose Standard......................................................................................................................................... 1449

Propriétés du tGSBucketList Standard ........................................................................................................................... 1447

Propriétés du tGSBucketExist Standard .......................................................................................................................... 1445

Scénario : Vider la mémoire avant d’y charger les données si une boucle existe dans le même sous-job.................. 1478

job.......................................................................................................................................................................................... 1478

Scénario 2 : Vider la mémoire avant d’y charger les données si une boucle existe dans le même sous-job.................. 1478

Scénario 1 : Lire des données directement dans la mémoire cache afin d’y accéder rapidement............... 1474

Propriétés du tHashInput Standard.................................................................................................................................1473

Scénario : Gérer des fichiers avec Google Cloud Storage ...................................................................................... 1465

Propriétés du tHashOutput Standard.............................................................................................................................1482

Scénarios associés...............................................................................................................................................................1484

Propriétés du tHBaseClose Standard.......................................................................................................................... 1485

Propriétés du tHBaseConnection Standard .................................................................................................................. 1487

Propriétés du tHBaseInput Standard.............................................................................................................................1493

Propriétés du tHBaseInput Standard.............................................................................................................................1493

Scénario : Echanger des données clients avec HBase............................................................................................ 1501
Scénario 1 : Envoyer une requête HTTP au serveur et sauvegarder localement les réponses HTTP

Scénarios associés

Propriétés du tHSQLDbRow Standard
Scénario associé

Propriétés du tHSQLDbInput Standard
Scénario associé

Connexion sécurisée à MapR
Scénarios associés

Propriétés du tHiveRow Standard
Scénario associé

Propriétés du tHiveLoad Standard
Scénario associé

Propriétés du tHiveInput Standard
Scénario associé

Propriétés du tHiveCreateTable Standard
Scénario associé

Propriétés du tHiveClose Standard
Scénario associé

Propriétés du tHDFSPut Standard
Scénario associé

Propriétés du tHDFSRename Standard
Scénario associé

Propriétés du tHDFSRowCount Standard
Scénario associé

Propriétés du tHDFSPut Standard
Scénario associé

Propriétés du tHDFSProperties Standard
Scénario associé

obtenues
Scénario 2 : Envoyer une requête POST depuis un fichier local JSON.......................................................... 1766

tImpalaClose......................................................................................................................................................... 1771
  Propriétés du tImpalaClose Standard.................................................................................................................. 1771
  Scénario associé..................................................................................................................................................... 1771

tImpalaConnection.................................................................................................................................................. 1774
  Propriétés du tImpalaConnection Standard........................................................................................................... 1774
  Scénario associé..................................................................................................................................................... 1777

tImpalaCreateTable.............................................................................................................................................. 1778
  Propriétés du tImpalaCreateTable Standard.......................................................................................................... 1778
  Scénario associé..................................................................................................................................................... 1784

tImpalaInput........................................................................................................................................................ 1785
  Propriétés du tImpalaInput Standard..................................................................................................................... 1785
  Scénario associé..................................................................................................................................................... 1790

tImpalaLoad........................................................................................................................................................ 1791
  Propriétés du tImpalaLoad Standard...................................................................................................................... 1791
  Scénario associé..................................................................................................................................................... 1796

tImpalaOutput...................................................................................................................................................... 1797
  Propriétés du tImpalaOutput Standard.................................................................................................................. 1797
  Scénario associé..................................................................................................................................................... 1802

tImpalaRow........................................................................................................................................................ 1803
  Propriétés du tImpalaRow Standard...................................................................................................................... 1803
  Scénarios associés.................................................................................................................................................. 1808

tInfiniteLoop..................................................................................................................................................... 1809
  Propriétés du tInfiniteLoop Standard.................................................................................................................... 1809
  Scénario associé..................................................................................................................................................... 1810

tInformixBulkExec............................................................................................................................................. 1811
  Propriétés du tInformixBulkExec Standard......................................................................................................... 1811
  Scénario associé..................................................................................................................................................... 1815

tInformixClose.................................................................................................................................................... 1817
  Propriétés du tInformixClose Standard............................................................................................................... 1817
  Scénario associé..................................................................................................................................................... 1818

tInformixCommit............................................................................................................................................... 1819
  Propriétés du tInformixCommit Standard.............................................................................................................. 1819
  Scénario associé..................................................................................................................................................... 1820

tInformixConnection......................................................................................................................................... 1821
  Propriétés du tInformixConnection Standard....................................................................................................... 1821
  Scénarios associés................................................................................................................................................ 1823

tInformixInput.................................................................................................................................................... 1824
  Propriétés du tInformixInput Standard................................................................................................................ 1824
  Scénarios associés............................................................................................................................................... 1827

tInformixOutput................................................................................................................................................ 1828
  Propriétés du tInformixOutput Standard............................................................................................................. 1828
  Scénarios associés............................................................................................................................................... 1833

tInformixOutputBulk................................................................................................................................. 1834
  Propriétés du tInformixOutputBulk Standard.................................................................................................... 1834
Scénarios associés......................................................................................................................................................................1837

**tInformixOutputBulkExec** ....................................................................................................................................................1838
  Propriétés du tInformixOutputBulkExec Standard........................................................................................................1838
  Scénarios associés....................................................................................................................................................................1842

**tInformixRow** ........................................................................................................................................................................1845
  Propriétés du tInformixRow Standard.................................................................................................................................1845
  Scénario associé........................................................................................................................................................................1849

**tInformixSCD** ..........................................................................................................................................................................1850
  Propriétés du tInformixSCD Standard................................................................................................................................1850
  Scénario associé........................................................................................................................................................................1853

**tInformixSP** ...........................................................................................................................................................................1854
  Propriétés du tInformixSP Standard..................................................................................................................................1854
  Scénarios associés....................................................................................................................................................................1857

**tingresBulkExec** ......................................................................................................................................................................1858
  Propriétés du tingresBulkExec Standard............................................................................................................................1858
  Scénario associé........................................................................................................................................................................1862

**tingresClose** ...........................................................................................................................................................................1863
  Propriétés du tingresClose Standard................................................................................................................................1863
  Scénario associé........................................................................................................................................................................1864

**tingresCommit** .........................................................................................................................................................................1865
  Propriétés du tingresCommit Standard.................................................................................................................................1865
  Scénario associé........................................................................................................................................................................1866

**tingresConnection** ...................................................................................................................................................................1867
  Propriétés du tingresConnection Standard...........................................................................................................................1867
  Scénario associé........................................................................................................................................................................1869

**tingresInput** ...........................................................................................................................................................................1870
  Propriétés du tingresInput Standard...................................................................................................................................1870
  Scénarios associés....................................................................................................................................................................1873

**tingresOutput** ........................................................................................................................................................................1874
  Propriétés du tingresOutput Standard................................................................................................................................1874
  Scénarios associés....................................................................................................................................................................1879

**tingresOutputBulk** .................................................................................................................................................................1880
  Propriétés du tingresOutputBulk Standard..........................................................................................................................1880
  Scénario associé........................................................................................................................................................................1882

**tingresOutputBulkExec** ..........................................................................................................................................................1883
  Propriétés du tingresOutputBulkExec Standard..................................................................................................................1883
  Scénario : Charger des données dans une table du SGBD Ingres.........................................................................................1886
  Scénarios associés....................................................................................................................................................................1889

**tingresRollback** ......................................................................................................................................................................1890
  Propriétés du tingresRollback Standard...............................................................................................................................1890
  Scénario associé........................................................................................................................................................................1891

**tingresRow** .............................................................................................................................................................................1892
Propriétés du tIngresRow Standard................................................................. 1892
Scénarios associés....................................................................................... 1895

tIngresSCD................................................................................................. 1896
Propriétés du tIngresSCD Standard............................................................... 1896
Scénario associé....................................................................................... 1899

tInterbaseClose....................................................................................... 1900
Propriétés du tInterbaseClose Standard..................................................... 1900
Scénario associé....................................................................................... 1901

tInterbaseCommit.................................................................................... 1902
Propriétés du tInterbaseCommit Standard................................................ 1902
Scénario associé....................................................................................... 1903

tInterbaseConnection.............................................................................. 1904
Propriétés du tInterbaseConnection Standard........................................... 1904
Scénarios associés................................................................................... 1905

tInterbaseInput....................................................................................... 1906
Propriétés du tInterbaseInput Standard..................................................... 1906
Scénarios associés................................................................................... 1909

tInterbaseOutput.................................................................................... 1910
Propriétés du tInterbaseOutput Standard................................................ 1910
Scénarios associés................................................................................... 1915

tInterbaseRollback................................................................................ 1916
Propriétés du tInterbaseRollback Standard............................................... 1916
Scénario associé....................................................................................... 1917

tInterbaseRow........................................................................................ 1918
Propriétés du tInterbaseRow Standard..................................................... 1918
Scénarios associés................................................................................... 1922

tIntervalMatch....................................................................................... 1923
Propriétés du tIntervalMatch Standard.................................................... 1923
Scénario : Identifier l'emplacement de serveurs à partir de leur IP............. 1924

tIterateToFlow....................................................................................... 1928
Propriétés du tIterateToFlow Standard.................................................... 1928
Scénario : Transformer une liste de fichiers en flux de données.................. 1929

tJasperOutput......................................................................................... 1932
Propriétés du tJasperOutput Standard....................................................... 1932
Scénario : Générer un rapport avec un modèle .jrxml............................... 1934

tJasperOutputExec............................................................................... 1937
Propriétés du tJasperOutputExec Standard............................................... 1937
Scénario associé....................................................................................... 1938

tJava........................................................................................................ 1939
Propriétés du tJava Standard..................................................................... 1939
Scénario : Imprimer le contenu d'une variable......................................... 1940

tJavaDBInput......................................................................................... 1944
Propriétés du tJavaDBInput Standard....................................................... 1944
Scénarios associés................................................................................... 1946

tJavaDBOutput....................................................................................... 1947
Suivre des modifications de données dans une table Snowflake à l'aide du composant tJDBCSCDELT.

**Propriétés du tJDBCSCDELT Standard**

Scénarios associés.

---

Scénario 1 : Générer un flux de données

**Propriétés du tJDBCRow Standard**

Scénario associé.

---

Scénario 2 : Traiter des lignes de données avec le tJavaFlex

**Propriétés du tJavaFlex Standard**

Scénario 1 associé.

---

Scénario : Transformer des données ligne par ligne avec un tJavaRow

**Propriétés du tJavaRow Standard**

---

Scénario 2 : Traiter des lignes de données avec le tJavaFlex

**Propriétés du tJavaFlex Standard**

---

Propriétés du tJavaDBRow Standard

Scénario associé.

---

Propriétés du tJavaDBOutput Standard

Scénario associé.

---

Propriétés du tJavaDBColumnList Standard

Scénario associé.

---

Propriétés du tJavaDBClose Standard

Scénario associé.

---

Propriétés du tJavaDBCommit Standard

Scénario associé.

---

Propriétés du tJavaDBConnection Standard

Scénario associé.

---

Propriétés du tJavaDBInput Standard

Scénarios associés.

---

Propriétés du tJavaDBOutput Standard

Scénarios associés.

---

Propriétés du tJavaDBRollback Standard

Scénario associé.

---

Propriétés du tJavaDBClose Standard

Scénario associé.

---

Propriétés du tJavaDBCommit Standard

Scénario associé.

---

Propriétés du tJavaDBColumnList Standard

Scénario associé.

---

Propriétés du tJavaDBClose Standard

Scénario associé.

---

Propriétés du tJavaDBConnection Standard

Scénario associé.

---

Propriétés du tJavaDBInput Standard

Scénarios associés.

---

Propriétés du tJavaDBOutput Standard

Scénarios associés.

---

Propriétés du tJavaDBRollback Standard

Scénario associé.

---

Propriétés du tJavaDBClose Standard

Scénario associé.

---

Propriétés du tJavaDBCommit Standard

Scénario associé.

---

Propriétés du tJavaDBColumnList Standard

Scénario associé.
Scénario : Afficher le contenu filtré d'un annuaire LDAP.

Propriétés du tLDAPConnection Standard.
Récupérer les informations de projet depuis l'application JIRA.

Scénario associé.

Propriétés du tLDAPClose Standard.
Mettre à jour un ticket dans l'application JIRA.

Scénario associé.

Propriétés du tLDAPAttributesInput Standard.

Scénario associé.

Propriétés du tKafkaOutput Standard.
Scénario associé.

Propriétés du tKafkaCreateTopic Standard.

Scénario associé.

Propriétés du tJoin Standard.

Scénario 1 : Faire une correspondance exacte entre deux colonnes et écrire les données rejetées.

Propriétés du tKafkaCommit Standard.

Scénario associé.

Propriétés du tKafkaConnection Standard.

Scénario associé.

Propriétés du tJMSOutput Standard.

Scénario associé.

Scénario : Mettre un message dans une file d’attente du serveur ActiveMQ et le retirer de cette file.

Mettre à jour un ticket dans l'application JIRA.

Créer un ticket dans l'application JIRA.

Récupérer les informations de projet depuis l'application JIRA.
Écrire des données relatives à des candidats dans une base de données MapR-DB OJAI

- **tLDAPRenameEntry**
  - Propriétés du tLDAPRenameEntry standard
  - Scénarios associés

- **tLibraryLoad**
  - Propriétés du tLibraryLoad standard
  - Scénario : Vérifier le format d’une adresse e-mail

- **tLineChart**
  - Propriétés du tLineChart standard
  - Scénario : Créer un graphique en lignes afin de faciliter l’analyse des tendances

- **tLogRow**
  - Propriétés du tLogRow standard
  - Scénarios associés

- **tLogCatcher**
  - Propriétés du tLogCatcher standard
  - Capturer les messages déclenchés par un composant tWarn
  - Capturer le message déclenché par un composant tDie

- **tLDAPRenameEntry**
  - Propriétés du tLDAPRenameEntry standard
  - Scénario : Editer des données dans un annuaire LDAP

- **tLibraryLoad**
  - Propriétés du tLibraryLoad standard
  - Scénario : Vérifier le format d’une adresse e-mail

- **tLineChart**
  - Propriétés du tLineChart standard
  - Scénario : Créer un graphique en lignes afin de faciliter l’analyse des tendances

- **tLogRow**
  - Propriétés du tLogRow standard
  - Scénarios associés

- **tLogCatcher**
  - Propriétés du tLogCatcher standard
  - Capturer les messages déclenchés par un composant tWarn
  - Capturer le message déclenché par un composant tDie

- **tLDAPRenameEntry**
  - Propriétés du tLDAPRenameEntry standard
  - Scénario : Editer des données dans un annuaire LDAP

- **tLibraryLoad**
  - Propriétés du tLibraryLoad standard
  - Scénario : Vérifier le format d’une adresse e-mail

- **tLineChart**
  - Propriétés du tLineChart standard
  - Scénario : Créer un graphique en lignes afin de faciliter l’analyse des tendances

- **tLogRow**
  - Propriétés du tLogRow standard
  - Scénarios associés

- **tLogCatcher**
  - Propriétés du tLogCatcher standard
  - Capturer les messages déclenchés par un composant tWarn
  - Capturer le message déclenché par un composant tDie

- **tLDAPRenameEntry**
  - Propriétés du tLDAPRenameEntry standard
  - Scénario : Editer des données dans un annuaire LDAP
Scénario : Exécuter une procédure stockée à l'aide du tMDMSP ........................................ 2320

Propriétés du tMDMSP Standard ..................................................................................................................2318

Scénario : Router un rapport de mise à jour vers l'Event Manager .....................................................2314

Propriétés du tMDMRouteRecord Standard ......................................................................................................... 2312

Scénario : Extraire des informations d'un enregistrement MDM au format XML ................................2305

Propriétés du tMDMReceive Standard ........................................................................................................................... 2303

Scénario : Charger des enregistrements dans une entité métier ...............................................................2291

Exemples d'opérations de mises à jour partielles à l'aide du tMDMOutput ........................................2285

Scénario : Écrire des données maître dans un hub MDM ................................................................. 2291

Supprimer partiellement des données maître du hub MDM .................................................................2296

Scénario : Charger des enregistrements dans une entité métier ...............................................................2251

Scénario associé ...................................................................................................................................................................... 2259

Propriétés du tMDMDelete Standard ..............................................................................................................................2262

Scénario associé ...................................................................................................................................................................... 2257

Propriétés du tMDMCommit Standard ........................................................................................................................... 2258

Scénario associé ...................................................................................................................................................................... 2256

Propriétés du tMDMClose Standard ..............................................................................................................................2256

Scénario associé ...................................................................................................................................................................... 2255

Propriétés du tMaxDBInput Standard ......................................................................................................................... 2236

Scénarios associés ..................................................................................................................................................................2238

Propriétés du tMaxDBOutput Standard ......................................................................................................................2239

Scénarios associés ..................................................................................................................................................................2243

Propriétés du tMaxDBRow Standard .............................................................................................................................. 2244

Scénario associé ...................................................................................................................................................................... 2244

Propriétés du tMDBBulkLoad Standard ...................................................................................................................... 2248

Scénario : Charger des enregistrements dans une entité métier ...............................................................2251

Propriétés du tMaxDBOutput Standard ......................................................................................................................2239

Scénarios associés ..................................................................................................................................................................2243

Propriétés du tMaxDBRow Standard .............................................................................................................................. 2244

Scénario associé ...................................................................................................................................................................... 2244

Propriétés du tMDBBulkLoad Standard ...................................................................................................................... 2248

Scénario : Charger des enregistrements dans une entité métier ...............................................................2251

Propriétés du tMaxDBInput Standard ......................................................................................................................... 2236

Scénarios associés ..................................................................................................................................................................2238

Propriétés du tMaxDBOutput Standard ......................................................................................................................2239

Scénarios associés ..................................................................................................................................................................2243

Propriétés du tMaxDBRow Standard .............................................................................................................................. 2244

Scénario associé ...................................................................................................................................................................... 2244

Propriétés du tMDBBulkLoad Standard ...................................................................................................................... 2248

Scénario : Charger des enregistrements dans une entité métier ...............................................................2251
Scénario : Échanger des informations concernant les événements d’un enregistrement MDM

**tMDMTriggerOutput**
- Propriétés du `tMDMTriggerOutput Standard
- Scénario associé

2336

**tMDMVieSearch**
- Propriétés du `tMDMVieSearch Standard
- Scénario : Récupérer des enregistrements d’un hub MDM via une vue existante

2339

**tMemorizeRows**
- Propriétés du `tMemorizeRows Standard
- Scénario : Récupérer les différents âges et l’âge le plus bas

2346

**tMemSQLClose**
- Propriétés du `tMemSQLClose Standard
- Scénario associé

2352

**tMemSQLConnection**
- Propriétés du `tMemSQLConnection Standard
- Scénario associé

2354

**tMemSQLInput**
- Propriétés du `tMemSQLInput Standard
- Scénario : Ecrire des données dans et lire des données d’une table d’une base de données MemSQL

2357

**tMemSQLOutput**
- Propriétés du `tMemSQLOutput Standard
- Scénario associé

2368

**tMemSQLRow**
- Propriétés du `tMemSQLRow Standard
- Scénario associé

2374

**tMicrosoftCrmInput**
- Propriétés du `tMicrosoftCrmInput Standard
- Scénario : Ecrire des données dans une base de données Microsoft CRM et attribuer des conditions aux colonnes pour extraire des lignes spécifiques

2379

**tMicrosoftCrmOutput**
- Propriétés du `tMicrosoftCrmOutput Standard
- Scénario associé

2390

**tMicrosoftMQInput**
- Propriétés du `tMicrosoftMQInput Standard
- Scénario : Ecrire et récupérer des messages dans une file d’attente de message Microsoft

2395

**tMicrosoftMQOutput**
- Propriétés du `tMicrosoftMQOutput Standard
- Scénario associé

2401

**tMomCommit**
- Propriétés du `tMomCommit Standard
- Scénario associé

2404

**tMomConnection**
- Propriétés du `tMomConnection Standard
- Scénario associé

2406

**tMomInput**
- Propriétés du `tMomInput Standard
- Scénario associé

2409
Scénario : Créer une collection et écrire des données dans celle-ci ...................................................................................................................... 2498
Scénario 2 : Effectuer une opération d’upsert sur des enregistrements dans une collection .................................................. 2503

tMongoDBRow .................................................................................................................................................................................. 2511
Propriétés du tMongoDBRow Standard ........................................................................................................................................ 2511
Scénario : Utiliser les fonctions MongoDB pour créer une collection et y écrire des données ........................................... 2515

tMSSqlInput .......................................................................................................................................................................................... 2520
Propriétés du tMSSqlInput Standard ........................................................................................................................................... 2520
Scénario associé .................................................................................................................................................................................. 2521

tMSSqlConnection ............................................................................................................................................................................. 2545
Propriétés du tMSSqlConnection Standard ........................................................................................................................................ 2545
Scénario associé .................................................................................................................................................................................. 2544

tMSSqlCommit .................................................................................................................................................................................. 2543
Propriétés du tMSSqlCommit Standard ........................................................................................................................................ 2543
Scénario associé .................................................................................................................................................................................. 2544

tMSSqlColumnList ............................................................................................................................................................................. 2540
Propriétés du tMSSqlColumnList Standard ........................................................................................................................................ 2540
Scénario associé .................................................................................................................................................................................. 2542

tMSSqlClose ..................................................................................................................................................................................... 2538
Propriétés du tMSSqlClose Standard ............................................................................................................................................... 2538
Scénario associé .................................................................................................................................................................................. 2539

tMSSqlBulkExec .............................................................................................................................................................................. 2533
Propriétés du tMSSqlBulkExec Standard ........................................................................................................................................ 2533
Scénarios associés ................................................................................................................................................................................ 2537

tMsgBox ......................................................................................................................................................................................... 2530
Propriétés du tMsgBox Standard ..................................................................................................................................................... 2530
Scénario : Test de type 'Hello world!' ............................................................................................................................................. 2531

tMSAXOutput .................................................................................................................................................................................. 2522
Propriétés du tMSAXOutput Standard ........................................................................................................................................... 2522
Scénario : Insérer des données dans une table définie sur le serveur Microsoft AX ............................................................. 2524
Scénario 2 : Effacer des données d’une table précise sur le serveur Microsoft AX ................................................................. 2526

tMSAXInput .................................................................................................................................................................................... 2520
Propriétés du tMSAXInput Standard ............................................................................................................................................... 2520
Scénario associé .................................................................................................................................................................................. 2521

Scénario : Utiliser les fonctions MongoDB pour créer une collection et y écrire des données ........................................... 2515

Scénarios associés................................................................................................................................................................................ 2576

Propriétés du tMSSqlOutputBulk Standard ........................................................................................................................................... 2569
Scénarios associés ................................................................................................................................................................................ 2571

Propriétés du tMSSqlOutputBulkExec Standard ...................................................................................................................................... 2572
Scénarios associés ................................................................................................................................................................................ 2576
Scénario : Écrire des colonnes d'une base de données MySQL dans un fichier de sortie en utilisant `tMysqlInput`...2635
Scénario : Écrire des colonnes d'une base de données MySQL dans un fichier de sortie en utilisant `tMysqlInput`...2635
Scénario : Utiliser des paramètres de contexte lors de la lecture d'une table d'une base de données...2638
Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte...2641
Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d'une base de données MySQL...2647

Scénario : Récupérer des informations personnelles à l'aide d'une procédure stockée...2594
Scénarios associés...2599

Propriétés du `tMSSqlTableList` Standard...2600
Scénario associé...2601

Propriétés du `tMSSqlRow` Standard...2579
Scénario associé...2578

Scénarios associés...2578

Propriétés du `tMSSqlSCD` Standard...2585
Scénarios associés...2584

Scénarios associés...2584

Propriétés du `tMSSqlRollback` Standard...2577
Scénario associé...2578

Propriétés du `tMSSqlLastInsertId` Standard...2579
Scénario associé...2578

Propriétés du `tMSSqlRow` Standard...2579
Scénario associé...2578

Scénario : Écrire des colonnes d'une base de données MySQL dans un fichier de sortie en utilisant `tMysqlInput`...2631
Scénario : Écrire des colonnes d'une base de données MySQL dans un fichier de sortie en utilisant `tMysqlInput`...2631
Scénario : Utiliser des paramètres de contexte lors de la lecture d'une table d'une base de données...2638
Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte...2641
Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d'une base de données MySQL...2647

Propriétés du `tMSSqlTableList` Standard...2600
Scénario associé...2601

Propriétés du `tMSSqlRow` Standard...2579
Scénario associé...2578

Scénarios associés...2578

Scénarios associés...2584

Scénarios associés...2584

Scénarios associés...2584
Scénario associé......................................................................................................................................................................2840
Propriétés du tNetezzaCommit Standard..............................................................................................................................2836
Écrire dans Neo4j des informations relatives à des acteurs et films avec une relation hiérarchique....2763
Scénario associé......................................................................................................................................................................2835
Propriétés du tNetezzaClose Standard.................................................................................................................................2834
Scénario associé......................................................................................................................................................................2833
Scénario 2 : Importer des données dans une base de données Neo4j à partir d’un fichier CSV à l’aide
de requête Cypher.................................................................................................................................................................2812
Scénario 1 : Créer des nœuds avec un libellé à l’aide d’une requête Cypher.................................................................2815
Scénario 2 : Importer des données dans une base de données Neo4j à partir d’un fichier CSV à l’aide
e de une requête Cypher.........................................................................................................................................................2819
Scénario 3 : Importer des données d’un fichier CSV dans Neo4j et créer des relations à l’aide d’une
requête Cypher......................................................................................................................................................................2825
Scénario associe......................................................................................................................................................................2832
Propriétés du tNetezzaImportTool Standard..........................................................................................................................2779
Propriétés du tNetezzaInput Standard.......................................................................................................................................2781
Scénario associe......................................................................................................................................................................2783
Propriétés du tNetezzaOutput Standard.......................................................................................................................................2784
Scénario 1 : Écrire des données dans une base de données Neo4j et lire des données spécifiques de
Cette base de données.............................................................................................................................................................2788
Scénario 2 : Écrire des informations concernant des familles dans Neo4j et créer des relations............2793
Scénario : Écrire des informations concernant des acteurs et des films dans Neo4j avec des relations
hiéarchiques..............................................................................................................................................................................2802
Scénario associé......................................................................................................................................................................2830
Propriétés du tNetezzaClose Standard.................................................................................................................................2834
Scénario associé......................................................................................................................................................................2835
Propriétés du tNetezzaCommit Standard.................................................................................................................................2836
Scénario associé......................................................................................................................................................................2837
Propriétés du tNetezzaConnection Standard..........................................................................................................................2838
Scénario associe......................................................................................................................................................................2840
<table>
<thead>
<tr>
<th>Class/Property</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>tNetezzaInput</td>
<td>2841</td>
</tr>
<tr>
<td>Propriétés du tNetezzaInput Standard</td>
<td>2841</td>
</tr>
<tr>
<td>Scénarios associés</td>
<td>2844</td>
</tr>
<tr>
<td>tNetezzaNzLoad</td>
<td>2845</td>
</tr>
<tr>
<td>Propriétés du tNetezzaNzLoad Standard</td>
<td>2845</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>2852</td>
</tr>
<tr>
<td>tNetezzaOutput</td>
<td>2853</td>
</tr>
<tr>
<td>Propriétés du tNetezzaOutput Standard</td>
<td>2853</td>
</tr>
<tr>
<td>Scénarios associés</td>
<td>2858</td>
</tr>
<tr>
<td>tNetezzaRollback</td>
<td>2859</td>
</tr>
<tr>
<td>Propriétés du tNetezzaRollback Standard</td>
<td>2859</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>2860</td>
</tr>
<tr>
<td>tNetezzaRow</td>
<td>2861</td>
</tr>
<tr>
<td>Propriétés du tNetezzaRow Standard</td>
<td>2861</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>2865</td>
</tr>
<tr>
<td>tNetezzaSCD</td>
<td>2866</td>
</tr>
<tr>
<td>Propriétés du tNetezzaSCD Standard</td>
<td>2866</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>2870</td>
</tr>
<tr>
<td>tNetsuiteConnection</td>
<td>2871</td>
</tr>
<tr>
<td>tNetsuiteConnection Standard properties</td>
<td>2871</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>2872</td>
</tr>
<tr>
<td>tNetsuitelInput</td>
<td>2873</td>
</tr>
<tr>
<td>Propriétés du tNetsuitelInput Standard</td>
<td>2873</td>
</tr>
<tr>
<td>Scénario : Gestion des données avec NetSuite</td>
<td>2876</td>
</tr>
<tr>
<td>tNetsuiteOutput</td>
<td>2882</td>
</tr>
<tr>
<td>Propriétés du tNetsuiteOutput Standard</td>
<td>2882</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>2885</td>
</tr>
<tr>
<td>tNormalize</td>
<td>2886</td>
</tr>
<tr>
<td>Propriétés du tNormalize Standard</td>
<td>2886</td>
</tr>
<tr>
<td>Scénario : Normaliser des données</td>
<td>2888</td>
</tr>
<tr>
<td>tOleDbInput</td>
<td>2891</td>
</tr>
<tr>
<td>Propriétés du tOleDbInput Standard</td>
<td>2891</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>2893</td>
</tr>
<tr>
<td>tOleDbOutput</td>
<td>2894</td>
</tr>
<tr>
<td>Propriétés du tOleDbOutput Standard</td>
<td>2894</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>2897</td>
</tr>
<tr>
<td>tOleDbRow</td>
<td>2898</td>
</tr>
<tr>
<td>Propriétés du tOleDbRow Standard</td>
<td>2898</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>2901</td>
</tr>
<tr>
<td>tOpenbravoERPInput</td>
<td>2902</td>
</tr>
<tr>
<td>Propriétés du tOpenbravoERPInput Standard</td>
<td>2902</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>2904</td>
</tr>
<tr>
<td>tOpenbravoERPOutput</td>
<td>2905</td>
</tr>
<tr>
<td>Propriétés du tOpenbravoERPOutput Standard</td>
<td>2905</td>
</tr>
<tr>
<td>Scénario associé</td>
<td>2907</td>
</tr>
</tbody>
</table>
Scénario associé : Vérifier le format de numéros à l'aide d'une procédure stockée

Propriétés du tOracleSCD Standard........................................................................................................2961
Scénario associé........................................................................................................................................2965

Scénario : Utiliser les paramètres de contexte lors de la lecture d’une table d’une base de données Oracle

Propriétés du tOracleRow Standard........................................................................................................2955
Scénario associé........................................................................................................................................2954

Scénario : Supprimer et insérer des données dans une base Oracle

Propriétés du tOracleBulkExec Standard..............................................................................................2908
Scénario associé........................................................................................................................................2914

tOracleClose..............................................................................................................................................2918

Propriétés du tOracleClose Standard........................................................................................................2918
Scénario associé........................................................................................................................................2919

tOracleCommit..........................................................................................................................................2920

Propriétés du tOracleCommit Standard.....................................................................................................2920
Scénario associé........................................................................................................................................2921

tOracleConnection...................................................................................................................................2922

Propriétés du tOracleConnection Standard.............................................................................................2922
Scénario associé........................................................................................................................................2925

tOracleInput................................................................................................................................................2926

Propriétés du tOracleInput Standard.........................................................................................................2926
Scénario associé........................................................................................................................................2925

tOracleOutput............................................................................................................................................2935

Propriétés du tOracleOutput Standard.......................................................................................................2935
Scénarios associés......................................................................................................................................2942

tOracleOutputBulk.................................................................................................................................2943

Propriétés du tOracleOutputBulk Standard............................................................................................2943
Scénarios associés......................................................................................................................................2945

tOracleOutputBulkExec............................................................................................................................2946

Propriétés du tOracleOutputBulkExec Standard......................................................................................2946
Scénarios associés......................................................................................................................................2945

tOracleRollback........................................................................................................................................2953

Propriétés du tOracleRollback Standard....................................................................................................2953
Scénario associé.........................................................................................................................................2954

tOracleRow................................................................................................................................................2955

Propriétés du tOracleRow Standard............................................................................................................2955
Scénario associé.........................................................................................................................................2954

tOracleSCD.................................................................................................................................................2961

Propriétés du tOracleSCD Standard............................................................................................................2961
Scénario associé.........................................................................................................................................2965

tOracleSCDELT..........................................................................................................................................2966

Propriétés du tOracleSCDELT Standard.....................................................................................................2966
Scénario associé.........................................................................................................................................2970

tOracleSP..................................................................................................................................................2971

Propriétés du tOracleSP Standard...............................................................................................................2971
Scénario : Vérifier le format de numéros à l’aide d’une procédure stockée.................................................2976
Scénarios associés......................................................................................................................................2979

tOracleTableList......................................................................................................................................2980

Propriétés du tOracleTableList Standard.................................................................................................2980
Scénario associé.........................................................................................................................................2981

tPaloCheckElements..............................................................................................................................2982
Scénario : Créer une règle dans un cube donné........................................................................................................3042
Propriétés du tPaloRule Standard................................................................................................................................... 3040
Scénario associé......................................................................................................................................................................3029

Scénario associé......................................................................................................................................................................3018

Propriétés du tPaloDimension Standard..........................................................................................................................3007
Scénario : Créer une dimension avec des éléments...........................................................................................................3010

Scénario asocié......................................................................................................................................................................2988
Propriétés du tPaloInputMulti Standard............................................................................................................................3020
Scénario : Récupérer des éléments de dimensions dans un cube donné...........................................................................3023

Scénario : Créer une base de données..............................................................................................................................2999
Propriétés du tPaloDatabase Standard..................................................................................................................................2999
Scénario associé......................................................................................................................................................................2988

Scénario associé......................................................................................................................................................................2986
Propriétés du tPaloCheckElements Standard......................................................................................................................2982
Scénario associé......................................................................................................................................................................2984

Scénario associe ......................................................................................................................................................................2986
Propriétés du tPaloConnection Standard.............................................................................................................................2987
Scénario associé......................................................................................................................................................................2986

Scénario : Créer un cube dans une base de données existante..........................................................................................2989
Propriétés du tPaloCube Standard.............................................................................................................................................2989

Découvrir le schéma de sortie en lecture seule du tPaloCubeList.......................................................................................2994
Propriétés du tPaloCubeList Standard..................................................................................................................................2994
Scénario : Récupérer des informations détaillées d’un cube d’une base de données déterminée........2996

Découvrir le schéma de sortie en lecture seule du tPaloDatabaseList..................................................................................3003
Propriétés du tPaloDatabaseList Standard..........................................................................................................................3003
Scénario : Récupérer des informations détaillées concernant les bases de données d’un serveur Palo donn...2996

Découvrir le schéma de sortie en lecture seule du tPaloCubeList.......................................................................................2994
Propriétés du tPaloCubeList Standard..................................................................................................................................2994
Scénario : Récupérer des informations détaillées concernant les dimensions d’une base de données déterminée...2996

Découvrir le schéma de sortie en lecture seule du tPaloDatabaseList..................................................................................3003
Propriétés du tPaloDatabaseList Standard..........................................................................................................................3003
Scénario : Récupérer des informations détaillées concernant les bases de données d’un serveur Palo donné....2996

Découvrir le schéma de sortie en lecture seule du tPaloDimensionList..................................................................................3015
Propriétés du tPaloDimensionList Standard..........................................................................................................................3015
Scénario : Récupérer des informations détaillées concernant les dimensions d’une base de données déterminée....3018

Propriétés du tPaloInputMulti Standard............................................................................................................................3020
Scénario : Récupérer des éléments de dimensions dans un cube donné...........................................................................3023

Propriétés du tPaloOutput Standard.........................................................................................................................................3027
Scénario associé......................................................................................................................................................................3029

Propriétés du tPaloClose Standard...........................................................................................................................................2985
Scénario associé......................................................................................................................................................................2984

Propriétés du tPaloOutputMulti Standard.............................................................................................................................3030
Scénario 1 : Ecrire des données dans un cube donné...............................................................................................................3033
Scénario 2 : Refuser un flux de données d’entrée lorsque les éléments à écrire n’existent pas dans un cube donné...3035

Propriétés du tPaloRule Standard.............................................................................................................................................3040
Scénario : Créer une règle dans un cube donné.......................................................................................................................3042

<table>
<thead>
<tr>
<th>Object</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>tPigAggregate</td>
<td>3091</td>
</tr>
<tr>
<td>tParseRecordSet</td>
<td>3089</td>
</tr>
<tr>
<td>tParAccelSCD</td>
<td>3085</td>
</tr>
<tr>
<td>tParAccelOutputBulkExec</td>
<td>3074</td>
</tr>
<tr>
<td>tParAccelOutputBulk</td>
<td>3071</td>
</tr>
<tr>
<td>tParAccelOutput</td>
<td>3065</td>
</tr>
<tr>
<td>tParAccelConnection</td>
<td>3058</td>
</tr>
<tr>
<td>tParAccelPrint</td>
<td>3061</td>
</tr>
<tr>
<td>tParAccelInput</td>
<td>3065</td>
</tr>
<tr>
<td>tParAccelOutputBulkExec</td>
<td>3074</td>
</tr>
<tr>
<td>tParAccelRollback</td>
<td>3078</td>
</tr>
<tr>
<td>tParAccelRow</td>
<td>3080</td>
</tr>
<tr>
<td>tParAccelSCD</td>
<td>3085</td>
</tr>
<tr>
<td>tParseRecordSet</td>
<td>3089</td>
</tr>
<tr>
<td>tPigAggregate</td>
<td>3091</td>
</tr>
<tr>
<td>tPigCode</td>
<td>3094</td>
</tr>
</tbody>
</table>

Découvrir le schéma de sortie en lecture seule du tPaloRuleList... 3044
Propriétés du tPaloRuleList Standard... 3044
Scénario : Récupérer des informations détaillées concernant des règles d'un cube donné... 3046

Propriétés du tParAccelBulkExec Standard... 3049
Scénarios associés... 3053
Propriétés du tParAccelClose Standard... 3054
Scénario associé... 3055
Propriétés du tParAccelCommit Standard... 3056
Scénario associé... 3057
Propriétés du tParAccelConnection Standard... 3058
Scénario associé... 3060
Propriétés du tParAccelInput Standard... 3061
Scénarios associés... 3064
Propriétés du tParAccelOutput Standard... 3065
Scénarios associés... 3070
Propriétés du tParAccelOutputBulk Standard... 3071
Scénarios associés... 3073
Propriétés du tParAccelOutputBulkExec Standard... 3074
Scénarios associés... 3077
Propriétés du tParAccelRollback Standard... 3078
Scénario associé... 3079
Propriétés du tParAccelRow Standard... 3080
Scénarios associés... 3084
Propriétés du tParAccelSCD Standard... 3085
Scénario associé... 3088
Propriétés du tParseRecordSet Standard... 3089
Scénario associé... 3090
Propriétés du tPigAggregate Standard... 3091
Scénario associé... 3093
Propriétés du tPigCode Standard... 3094
Scénario : Sélectionner une colonne de données d’un fichier d’entrée et la stocker dans un fichier local 3096

tPigCoGroup..................................................................................................................................................3100
Propriétés du tPigCoGroup Standard........................................................................................................3100
Scénario associé..................................................................................................................................................3102

tPigCross..................................................................................................................................................3110
Propriétés du tPigCross Standard..................................................................................................................3110
Scénario associé..................................................................................................................................................3112

tPigDistinct...............................................................................................................................................3113
Propriétés du tPigDistinct Standard..............................................................................................................3113
Scénario associé..................................................................................................................................................3115

tPigFilterColumns........................................................................................................................................3116
Propriétés du tPigFilterColumns Standard.....................................................................................................3116
Scénario associé..................................................................................................................................................3118

tPigFilterRow...............................................................................................................................................3119
Propriétés du tPigFilterRow Standard............................................................................................................3119
Scénario : Filtrer des lignes de données selon une condition et sauvegarder le résultat dans un fichier local........................................................................................................................................3121

tPigJoin..................................................................................................................................................3125
Propriétés du tPigJoin Standard......................................................................................................................3125
Scénario : Effectuer une jointure sur deux fichiers à partir d’une correspondance exacte et sauvegarder le résultat dans un fichier local......................................................................................................................................3128

tPigLoad..................................................................................................................................................3135
Propriétés du tPigLoad Standard.....................................................................................................................3135
Scénario : Charger une table HBase..............................................................................................................3135

tPigMap..................................................................................................................................................3155
Paramètres de mapping facultatifs..................................................................................................................3155
Propriétés du tPigMap Standard......................................................................................................................3156
Scénario : Effectuer une jointure dans un processus Pig sur des données concernant les conditions de circulation........................................................................................................................................3157

tPigReplicate........................................................................................................................................3165
Propriétés du tPigReplicate Standard.............................................................................................................3165
Scénario : Répliquer un flux et trier deux flux identiques..............................................................................3167

tPigSort..................................................................................................................................................3175
Propriétés du tPigSort Standard......................................................................................................................3175
Scénario : Trier des données en ordre ascendant..........................................................................................3177

tPigStoreResult........................................................................................................................................3181
Propriétés du tPigStoreResult Standard.........................................................................................................3181
Scénario associé..................................................................................................................................................3188

tPivotToColumnsDelimited..........................................................................................................................3189
Propriétés du tPivotToColumnsDelimited Standard......................................................................................3189
Scénario : Utiliser une colonne pivot pour agréger des données.................................................................3190

tPOP..................................................................................................................................................3193
Propriétés du tPOP Standard..........................................................................................................................3193
Scénario : Récupérer une sélection d’e-mails à partir d’un compte de messagerie......................................3195

tPostgresPlusBulkExec..............................................................................................................................3198
tPostgresqlClose.......................................................................................................................................................... 3202
Propriétés du tPostgresqlClose Standard.................................................................................................................. 3202
Scénario associé.......................................................................................................................................................... 3203

tPostgresqlCommit....................................................................................................................................................... 3204
Propriétés du tPostgresqlCommit Standard.................................................................................................................. 3204
Scénario associé.......................................................................................................................................................... 3205

tPostgresPlusSCDELT.................................................................................................................................................... 3206
Propriétés du tPostgresPlusSCD Standard.................................................................................................................... 3206
Scénario associé.......................................................................................................................................................... 3207

tPostgresPlusRow.......................................................................................................................................................... 3209
Propriétés du tPostgresPlusRow Standard.................................................................................................................... 3209
Scénarios associés....................................................................................................................................................... 3212

tPostgresPlusRollback.................................................................................................................................................. 3213
Propriétés du tPostgresPlusRollback Standard............................................................................................................ 3213
Scénarios associés....................................................................................................................................................... 3218

tPostgresPlusOutputBulkExec....................................................................................................................................... 3219
Propriétés du tPostgresPlusOutputBulkExec Standard.................................................................................................. 3219
Scénarios associés....................................................................................................................................................... 3221

tPostgresPlusOutput..................................................................................................................................................... 3222
Propriétés du tPostgresPlusOutput Standard................................................................................................................ 3222
Scénarios associés....................................................................................................................................................... 3225

tPostgresPlusBulkExec................................................................................................................................................ 3228
Propriétés du tPostgresPlusBulkExec Standard........................................................................................................ 3228
Scénarios associés....................................................................................................................................................... 3232

tPostgresPlusSCD.......................................................................................................................................................... 3233
Propriétés du tPostgresPlusSCD Standard.................................................................................................................... 3233
Scénario associé.......................................................................................................................................................... 3236

tPostgresPlusSCDELT.................................................................................................................................................. 3237
Propriétés du tPostgresPlusSCDELT Standard................................................................................................................ 3237
Scénario associé.......................................................................................................................................................... 3241

tPostgresqlBulkExec.................................................................................................................................................. 3242
Propriétés du tPostgresqlBulkExec Standard................................................................................................................ 3242
Scénarios associés....................................................................................................................................................... 3246

tPostgresqlClose.......................................................................................................................................................... 3247
Propriétés du tPostgresqlClose Standard...................................................................................................................... 3247
Scénario associé.......................................................................................................................................................... 3248

tPostgresqlCommit....................................................................................................................................................... 3249
Propriétés du tPostgresqlCommit Standard................................................................................................................... 3249
Scénario associé.......................................................................................................................................................... 3250

tPostgresqlConnection.................................................................................................................................................. 3251
Scénario associé.................................................................................................................................................. 3318
tRedshiftInput.................................................................................................................................................. 3319
  Propriétés du tRedshiftInput Standard........................................................................................................... 3319
  Scénario : Manipuler des données avec Redshift.......................................................................................... 3323
tRedshiftOutput................................................................................................................................................. 3329
  Propriétés du tRedshiftOutput Standard.......................................................................................................... 3329
  Scénarios associés........................................................................................................................................... 3335
tRedshiftOutputBulk......................................................................................................................................... 3336
  Propriétés du tRedshiftOutputBulk Standard.................................................................................................. 3336
  Scénario associé........................................................................................................................................... 3339
tRedshiftOutputBulkExec.................................................................................................................................. 3340
  Propriétés du tRedshiftOutputBulkExec Standard.......................................................................................... 3340
  Scénario associé........................................................................................................................................... 3345
tRedshiftRollback.............................................................................................................................................. 3346
  Propriétés du tRedshiftRollback Standard....................................................................................................... 3346
  Scénario associé........................................................................................................................................... 3347
tRedshiftRow.................................................................................................................................................... 3348
  Propriétés du tRedshiftRow Standard.................................................................................................................. 3348
  Scénarios associés........................................................................................................................................... 3352
tRedshiftUnload................................................................................................................................................ 3353
  Propriétés du tRedshiftUnload Standard........................................................................................................... 3353
  Scénario associé........................................................................................................................................... 3357
tReplace........................................................................................................................................................... 3358
  Propriétés du tReplace Standard.......................................................................................................................... 3358
  Scénario : Remplacement multiple et filtrage de colonnes.............................................................................. 3360
tReplaceList..................................................................................................................................................... 3363
  Propriétés du tReplaceList Standard.................................................................................................................. 3363
  Scénario : Remplacement à partir d'un fichier de référence............................................................................ 3364
tReplicate......................................................................................................................................................... 3368
  Propriétés du tReplicate Standard...................................................................................................................... 3368
  Scénario : Répliquer un flux et trier deux flux identiques.................................................................................. 3369
tREST............................................................................................................................................................... 3374
  Propriétés du tREST Standard........................................................................................................................... 3374
  Scénario : Créer et récupérer des données en invoquant un service Web REST.............................................. 3376
tRESTClient....................................................................................................................................................... 3379
  Propriétés du tRESTClient Standard.................................................................................................................. 3379
  Scénario 1 : Obtenir des informations sur un utilisateur en interagissant avec un service RESTful................ 3385
  Scénario 2 : Mettre à jour les informations des utilisateurs via une interaction avec un service RESTful........ 3392
tRESTRequest.................................................................................................................................................... 3400
  Propriétés du tRESTRequest Standard.............................................................................................................. 3400
  Scénario 1 : Service REST acceptant des requêtes HTTP GET et envoyant des réponses........................... 3404
  Scénario 2 : Utiliser les paramètres URI Query pour explorer les données d'une base de données................ 3409
  Scénario 3 : Service REST acceptant des requêtes HTTP POST ................................................................. 3418
  Scénario 4 : Service REST acceptant les requêtes POST HTTP et envoyant des réponses........................ 3424
  Scénario 5 : Service REST acceptant des requêtes POST HTTP dans un formulaire HTML........................ 3433
Exécuter dynamiquement une liste de Jobs fils .........................................................................................................3506

Propriétés du tRunJob Standard.......................................................................................................................................3498
Scénario associé..............................................................................................................................................................3499

Scénario 3 : Créer un fichier XML de flux ATOM ......................................................................................................3495

Scénario 2 : Créer un flux RSS contenant des métadonnées ................................................................................3492

Propriétés du tRSSOutput Standard................................................................................................................................3486
Scénario associé..............................................................................................................................................................3487

Scénario : Récupérer les billets fréquemment mis à jour sur un blog .............................................................. 3484

Propriétés du tRSSInput Standard ...................................................................................................................................3483
Scénario associé..............................................................................................................................................................3484

Scénario : Génération aléatoire de fichiers de test .................................................................................................. 3481

Propriétés du tRowGenerator Standard.........................................................................................................................3478
Scénario associé..............................................................................................................................................................3479

Scénario associé......................................................................................................................................................................3478

Propriétés du tRowGenerator Standard.........................................................................................................................3478
Scénario associé..............................................................................................................................................................3479

Propriétés du tRiakKeyList Standard ..............................................................................................................................3455
Scénario associé..............................................................................................................................................................3456

Scénario associé......................................................................................................................................................................3455

Propriétés du tRiakKeyList Standard ..............................................................................................................................3455
Scénario associé..............................................................................................................................................................3456

Propriétés du tRiakOutput Standard..................................................................................................................................3457
Scénario associé..............................................................................................................................................................3458

Propriétés du tRiakInput Standard ....................................................................................................................................3449
Scénario : Exporter des données depuis un bucket Riak vers un fichier local.......................................................... 3451

Propriétés du tRiakBucketList Standard ........................................................................................................................3443
Scénario associé..............................................................................................................................................................3444

Scénario associé......................................................................................................................................................................3443

Propriétés du tRiakBucketList Standard ........................................................................................................................3443
Scénario associé..............................................................................................................................................................3444

Propriétés du tRiakConnection Standard ......................................................................................................................3447
Scénario associé..............................................................................................................................................................3448

Scénario associé......................................................................................................................................................................3447

Propriétés du tRiakConnection Standard ......................................................................................................................3447
Scénario associé..............................................................................................................................................................3448

Propriétés du tRiakInput Standard ....................................................................................................................................3449
Scénario : Échanger des messages entre un Job et une Route... 3462

Propriétés du tRouteFault Standard............................................................................................................................... 3460
Scénario : Échanger des messages entre un Job et une Route... 3462

Propriétés du tRouteInput Standard............................................................................................................................... 3468
Scénario : Échanger des messages entre un Job et une Route... 3469

Propriétés du tRouteOutput Standard...........................................................................................................................3475
Scénario associé..............................................................................................................................................................3477

Propriétés du tRowGenerator Standard.........................................................................................................................3478
Scénario associé..............................................................................................................................................................3479

Propriétés du tRSSInput Standard....................................................................................................................................3483
Scénario : Récupérer les billets fréquemment mis à jour sur un blog... 3484

Propriétés du tRSSOutput Standard...............................................................................................................................3486
Scénario 1 : Créer un flux RSS et stocker les fichiers sur un serveur FTP... 3488
Scénario 2 : Créer un flux RSS contenant des métadonnées... 3492
Scénario 3 : Créer un fichier XML de flux ATOM... 3495

Propriétés du tRunJob Standard...........................................................................................................................................3498
Appeler un Job et passer le paramètre nécessaire au Job appelé... 3501
Exécuter dynamiquement une liste de Jobs fils... 3506
Propager les données de sortie en mémoire tampon du Job fils au Job père..........................3509

**tS3BucketCreate**..................................................................................................................3513
Propriétés du tS3BucketCreate Standard............................................................................3513
Scénario associé......................................................................................................................3515

**tS3BucketDelete**..................................................................................................................3516
Propriétés du tS3BucketDelete Standard..............................................................................3516
Scénario associé......................................................................................................................3518

**tS3BucketExist**.................................................................................................................3519
Propriétés du tS3BucketExist Standard...............................................................................3519
Scénario : Vérifier l'absence d'un bucket, le créer et lister tous les buckets S3..............3521

**tS3BucketList**.....................................................................................................................3526
Propriétés du tS3BucketList Standard..................................................................................3526
Scénario associé......................................................................................................................3528

**tS3Close**..................................................................................................................................3529
Propriétés du tS3Close Standard..........................................................................................3529
Scénario associé......................................................................................................................3530

**tS3Connection**..................................................................................................................3531
Propriétés du tS3Connection Standard.................................................................................3531
Scénario associé......................................................................................................................3533

**tS3Copy**...............................................................................................................................3534
Propriétés du tS3Copy Standard...........................................................................................3534
Scénario : Copier un objet S3 d'un bucket à un autre........................................................3536

**tS3Delete**............................................................................................................................3541
Propriétés du tS3Delete Standard.........................................................................................3541
Scénario associé......................................................................................................................3543

**tS3Get**....................................................................................................................................3544
Propriétés du tS3Get Standard.............................................................................................3544
Scénario associé......................................................................................................................3547

**tS3List**...................................................................................................................................3548
Propriétés du tS3List Standard.............................................................................................3548
Scénario : Lister les fichiers d'un bucket ayant le même préfixe........................................3550

**tS3Put**....................................................................................................................................3555
Propriétés du tS3Put Standard...............................................................................................3555
Scénario : Échange de fichiers avec Amazon S3.................................................................3558

**tSageX3Input**.....................................................................................................................3562
Propriétés du tSageX3Input Standard...................................................................................3562
Scénario : Utiliser les clés des requêtes pour extraire des données d'un système Sage X3 donné......3565

**tSageX3Output**..................................................................................................................3569
Propriétés du tSageX3Output Standard................................................................................3569
Scénario : Utiliser un service Web Sage X3 pour insérer des données dans un système Sage X3 donné.3571

**tSalesforceBulkExec**.......................................................................................................3575
Propriétés du tSalesforceBulkExec Standard......................................................................3575
Scénario associé......................................................................................................................3579

**tSalesforceConnection**...................................................................................................3580
Propriétés du tSalesforceConnection Standard......................................................................3580
Se connecter à Salesforce à l'aide d'un flux implicite OAuth pour authentifier l'utilisateur (déprécié)... 3583
Scénario associé ................................................................................................................................................... 3588

tSAPHanaClose .................................................................................................................................................. 3662
Propriétés du tSAPHanaClose Standard ........................................................................................................... 3662
Scénario associé ................................................................................................................................................... 3662

SalesforceWaveOutputBulkExec ................................................................................................................... 3653
Propriétés du tSalesforceWaveOutputBulkExec Standard .............................................................................. 3653
Scénario associé ................................................................................................................................................... 3653

SalesforceWaveBulkExec ................................................................................................................................. 3648
Propriétés du tSalesforceWaveBulkExec Standard ........................................................................................... 3648
Scénario associé ................................................................................................................................................... 3648

SalesforceOutputBulk ......................................................................................................................................... 3640
Propriétés du tSalesforceOutputBulk Standard .............................................................................................. 3640
Scénario associé ................................................................................................................................................... 3640

SalesforceInput .................................................................................................................................................... 3607
Propriétés du tSalesforceInput Standard ............................................................................................................ 3607
Scénario associé ................................................................................................................................................... 3607

SalesforceGetUpdated ......................................................................................................................................... 3602
Propriétés du tSalesforceGetUpdated Standard .............................................................................................. 3602
Scénario associé ................................................................................................................................................... 3602

SalesforceOutput .................................................................................................................................................. 3619
Propriétés du tSalesforceOutput Standard ......................................................................................................... 3619
Effectuer un upsert sur des données Salesforce en se basant sur des ID externes ............................................. 3625

SalesforceOutputBulk ......................................................................................................................................... 3636
Propriétés du tSalesforceOutputBulk Standard .............................................................................................. 3636
Scénario associé ................................................................................................................................................... 3636

SalesforceOutputBulkExec ................................................................................................................................ 3639
Propriétés du tSalesforceOutputBulkExec Standard .......................................................................................... 3639

SalesforceWaveBulkExec .................................................................................................................................... 3648
Propriétés du tSalesforceWaveBulkExec Standard ............................................................................................ 3648
Scénario associé ................................................................................................................................................... 3648

SalesforceWaveOutputBulkExec .......................................................................................................................... 3653
Propriétés du tSalesforceWaveOutputBulkExec Standard .................................................................................... 3653
Scénario associé ................................................................................................................................................... 3653

SampleRow .......................................................................................................................................................... 3658
Propriétés du tSampleRow Standard ................................................................................................................... 3658
Scénario : Filtrer des lignes et des groupes de lignes ....................................................................................... 3659

SAPHanaClose ....................................................................................................................................................... 3662
Propriétés du tSAPHanaClose Standard ............................................................................................................. 3662
Scénario associé ................................................................................................................................................... 3662

SAPHanaCommit .................................................................................................................................................... 3663
Propriétés du tSAPHanaCommit Standard ........................................................................................................... 3663
Scénario associé ................................................................................................................................................... 3663

SAPHanaConnection ............................................................................................................................................ 3665
Propriétés du tSAPHanaConnection Standard ................................................................................................. 3665
Scénario associé ................................................................................................................................................... 3665

SAPHanaInput ........................................................................................................................................................ 3667
Propriétés du tSAPHanaInput Standard .............................................................................................................. 3667
Scénario associé ................................................................................................................................................... 3667
tSAPHanaOutput.................................................................................................................................3671
Propriétés du tSAPHanaOutput Standard.............................................................................................3671
Scénario associé.................................................................................................................................3676

**tSAPHanaRollback**............................................................................................................................3677
Propriétés du tSAPHanaRollback Standard..........................................................................................3677
Scénario associé.................................................................................................................................3678

**tSAPHanaRow**.....................................................................................................................................3679
Propriétés du tSAPHanaRow Standard....................................................................................................3679
Scénario associé......................................................................................................................................3682

**tSasInput**..........................................................................................................................................3683
Propriétés du tSasInput Standard............................................................................................................3683
Scénarios associés.................................................................................................................................3685

**tSasOutput**........................................................................................................................................3686
Propriétés du tSasOutput Standard........................................................................................................3686
Scénarios associés.................................................................................................................................3690

**tSchemaComplianceCheck**.............................................................................................................3691

**tSCPClose**.........................................................................................................................................3692
Propriétés du tSCPClose Standard.........................................................................................................3692
Scénario associé......................................................................................................................................3693

**tSCPConnection**..............................................................................................................................3694
Propriétés du tSCPConnection Standard..............................................................................................3694
Scénarios associés.................................................................................................................................3695

**tSCPDelete**.........................................................................................................................................3696
Propriétés du tSCPDelete Standard........................................................................................................3696
Scénario associé......................................................................................................................................3697

**tSCPFileExists**.................................................................................................................................3698
Propriétés du tSCPFileExists Standard..................................................................................................3698
Scénario : Gérer un fichier en utilisant SCP.........................................................................................3699

**tSCPFileList**.......................................................................................................................................3705
Propriétés du tSCPFileList Standard.....................................................................................................3705
Scénario associé......................................................................................................................................3707

**tCPGet**...............................................................................................................................................3708
Propriétés du tCPGet Standard.............................................................................................................3708
Scénario associé......................................................................................................................................3709

**tSCPPut**...............................................................................................................................................3710
Propriétés du tSCPPut Standard.............................................................................................................3710
Scénario associé......................................................................................................................................3711

**tSCPRename**.......................................................................................................................................3712
Propriétés du tSCPRename Standard.....................................................................................................3712
Scénario associé......................................................................................................................................3713

**tSCPTruncate**.....................................................................................................................................3714
Propriétés du tSCPTruncate Standard....................................................................................................3714
Scénario associé......................................................................................................................................3715

**tSendMail**.........................................................................................................................................3716
Propriétés du tSendMail Standard..........................................................................................................3716
Scénario : Envoyer un e-mail lors d'une erreur.........................................................................................................................3718

**tServerAlive** .................................................................................................................................................................................3722
Propriétés du tServerAlive Standard...............................................................................................................................................3722
Scénario associé................................................................................................................................................................................3723

**tServiceNowConnection** ..............................................................................................................................................................3726
Propriétés du tServiceNowConnection Standard........................................................................................................................3726
Scénario associé................................................................................................................................................................................3727

**tServiceNowInput** .........................................................................................................................................................................3728
Propriétés du tServiceNowInput Standard..................................................................................................................................3728
Scénario associé................................................................................................................................................................................3731

**tServiceNowOutput** .......................................................................................................................................................................3732
Propriétés du tServiceNowOutput Standard.................................................................................................................................3732
Scénario associé................................................................................................................................................................................3735

**tSetEnv** ..........................................................................................................................................................................................3736
Propriétés du tSetEnv Standard........................................................................................................................................................3736
Scénario : Modifier une variable lors de l'exécution d'un Job........................................................................................................3737

**tSetGlobalVar** .................................................................................................................................................................................3740
Propriétés du tSetGlobalVar Standard........................................................................................................................................3740
Scénario : Afficher le contenu d'une variable globale......................................................................................................................3741

**tSetKerberosConfiguration** ..........................................................................................................................................................3743
Propriétés du tSetKerberosConfiguration Standard.......................................................................................................................3743
Scénario associé................................................................................................................................................................................3744

**tSetKeystore** ...............................................................................................................................................................................3745
Propriétés du tSetKeystore Standard........................................................................................................................................3745
Scénario : Extraire des informations client d'un fichier WSDL privé.............................................................................................3746

**tSetProxy** .....................................................................................................................................................................................3751
Propriétés du tSetProxy Standard................................................................................................................................................3751
Scénario associé................................................................................................................................................................................3752

**tSleep** ..........................................................................................................................................................................................3753
Propriétés du tSleep Standard........................................................................................................................................................3753
Scénario associé................................................................................................................................................................................3754

**tSnowflakeClose** ............................................................................................................................................................................3755
Propriétés du tSnowflakeClose Standard......................................................................................................................................3755
Scénario associé................................................................................................................................................................................3756

**tSnowflakeConnection** .................................................................................................................................................................3757
Propriétés du tSnowflakeConnection Standard................................................................................................................................3757
Scénario associé................................................................................................................................................................................3759

**tSnowflakeInput** ............................................................................................................................................................................3760
Propriétés du tSnowflakeInput Standard......................................................................................................................................3760
Scénario : Écrire et lire des données dans une table Snowflake....................................................................................................3763

**tSnowflakeOutput** .........................................................................................................................................................................3769
Propriétés du tSnowflakeOutput Standard......................................................................................................................................3769
Scénario associé................................................................................................................................................................................3772

**tSnowflakeRow** ............................................................................................................................................................................3773
Propriétés du tSnowflakeRow Standard........................................................................................................................................3773
Scénario associé ...................................................................................................................................................................... 3777

tSOAP............................................................................................................................................................................. 3778
Propriétés du tSOAP Standard ............................................................................................................................................. 3778
Scénario : Récupérer le nom d'un pays en utilisant un Service Web .................................................................................. 3780
Scénario 2 : Utiliser un message SOAP depuis un fichier XML pour obtenir le nom d'un pays et le sauvegarder dans un fichier XML .................................................................................................................. 3782

tSocketInput ...................................................................................................................................................................... 3786
Propriétés du tSocketInput Standard ...................................................................................................................................... 3786
Scénario : Transférer des données vers un port d'écoute ................................................................................................. 3788

tSocketOutput ..................................................................................................................................................................... 3791
Propriétés du tSocketOutput Standard .................................................................................................................................. 3791
Scénario associé ...................................................................................................................................................................... 3793

tSortRow ........................................................................................................................................................................... 3794
Propriétés du tSortRow Standard .......................................................................................................................................... 3794
Trier des données .................................................................................................................................................................. 3796

tSplitRow ........................................................................................................................................................................... 3798
Propriétés du tSplitRow Standard .......................................................................................................................................... 3798
Scénario 1 : Séparer une ligne en plusieurs lignes ............................................................................................................. 3799

tSplunkEventCollector .......................................................................................................................................................... 3803
Propriétés du tSplunkEventCollector Standard ................................................................................................................. 3803
Scénario associé ...................................................................................................................................................................... 3805

tSPSSInput .......................................................................................................................................................................... 3806
Propriétés du tSPSSInput Standard ........................................................................................................................................... 3806
Scénario : Afficher le contenu d'un fichier SPSS .................................................................................................................. 3808

tSPSSOutput ........................................................................................................................................................................ 3811
Propriétés du tSPSSOutput Standard ..................................................................................................................................... 3811
Scénario : Ecrire des données dans un fichier .sav ............................................................................................................ 3813

tSPSSProperties ................................................................................................................................................................. 3816
Propriétés du tSPSSProperties Standard ............................................................................................................................ 3816
Scénario associé ...................................................................................................................................................................... 3817

tSPSSStructure ................................................................................................................................................................. 3818
Propriétés du tSPSSStructure Standard ............................................................................................................................. 3818
Scénario associé ...................................................................................................................................................................... 3819

tSQLDWHBulkExec ............................................................................................................................................................... 3821
Propriétés du tSQLDWHBulkExec Standard .......................................................................................................................... 3821
Scénario associé ...................................................................................................................................................................... 3826

tSQLDWHClose .................................................................................................................................................................... 3827
Propriétés du tSQLDWHClose Standard ................................................................................................................................. 3827
Scénario associé ...................................................................................................................................................................... 3828

tSQLDWHCommit ............................................................................................................................................................... 3829
Propriétés du tSQLDWHCommit Standard ............................................................................................................................ 3829
Scénario associé ...................................................................................................................................................................... 3830

tSQLDWHConnection ............................................................................................................................................................ 3831
Propriétés du tSQLDWHConnection Standard .................................................................................................................... 3831
Scénario associé ...................................................................................................................................................................... 3833
Propriétés du `tSQLTemplateFilterColumns` Standard ...............................................................................................3890
Scénario associé ..................................................................................................................................................................3893

Propriétés du `tSQLTemplateCommit` Standard ...........................................................................................................3888
Scénario associé ..................................................................................................................................................................3891

Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD ..........................................................3883

Propriétés du `tSQLTemplateAggregate` Standard ......................................................................................................3881
Scénario associé ..................................................................................................................................................................3884

Propriétés du `tSQLTemplate` Standard ..........................................................................................................................3878
Scénario associé ..................................................................................................................................................................3881

Scénario associé ..................................................................................................................................................................3877

Scénario : Mettre à jour des lignes SQLite ..................................................................................................................3875

Propriétés du `tSQLiteRow` Standard ...............................................................................................................................3872
Scénario associé ..................................................................................................................................................................3875

Propriétés du `tSQLiteRollback` Standard .......................................................................................................................3870
Scénario associé ..................................................................................................................................................................3873

Propriétés du `tSQLiteOutput` Standard ..........................................................................................................................3864
Scénario associé ..................................................................................................................................................................3867

Propriétés du `tSQLiteConnection` Standard .................................................................................................................3856
Scénario associé ..................................................................................................................................................................3859

Propriétés du `tSQLiteCommit` Standard ........................................................................................................................3854
Scénario associé ..................................................................................................................................................................3857

Propriétés du `tSQLiteClose` Standard ............................................................................................................................3852
Scénario associé ..................................................................................................................................................................3855

Scénario associé ..................................................................................................................................................................3850

Scénario associé ..................................................................................................................................................................3846

Propriétés du `tSQLDWHRollback` Standard ..................................................................................................................3845
Scénario associé ..................................................................................................................................................................3848

Propriétés du `tSQLDWHOutput` Standard .....................................................................................................................3838
Scénario associé ..................................................................................................................................................................3841

Propriétés du `tSQLDWHInput` Standard .........................................................................................................................3834
Scénario associé ..................................................................................................................................................................3837

Propriétés du `tSQLDWHRow` Standard ...........................................................................................................................3847
Scénario associé ..................................................................................................................................................................3850

Propriétés du `tSQLiteClose` Standard ............................................................................................................................3852
Scénario associé ..................................................................................................................................................................3855

Propriétés du `tSQLiteCommit` Standard ........................................................................................................................3854
Scénario associé ..................................................................................................................................................................3857

Propriétés du `tSQLiteClose` Standard ............................................................................................................................3852
Scénario associé ..................................................................................................................................................................3855

Propriétés du `tSQLiteConnection` Standard .................................................................................................................3856
Scénario associé ..................................................................................................................................................................3859

Propriétés du `tSQLiteCommit` Standard ........................................................................................................................3854
Scénario associé ..................................................................................................................................................................3857

Propriétés du `tSQLiteClose` Standard ............................................................................................................................3852
Scénario associé ..................................................................................................................................................................3855

Propriétés du `tSQLTemplateAggregate` Standard ........................................................................................................3881
Scénario : Filtrer et agréger les colonnes d'une table directement dans le SGBD ..........................................................3884

Propriétés du `tSQLTemplateCommit` Standard .............................................................................................................3888
Scénario associé ..................................................................................................................................................................3890

Propriétés du `tSQLTemplateFilterColumns` Standard ..................................................................................................3890
<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>3919</td>
<td>tSqoopImport</td>
<td>Propriétés du tSqoopImport Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénario : Importer une table MySQL dans HDFS</td>
</tr>
<tr>
<td>3936</td>
<td>tSqoopImportAllTables</td>
<td>Propriétés du tSqoopImportAllTables Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénario associé</td>
</tr>
<tr>
<td>3946</td>
<td>tSqoopMerge</td>
<td>Propriétés du tSqoopMerge Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénario : Fusionner deux ensembles de données dans HDFS</td>
</tr>
<tr>
<td>3960</td>
<td>tSQSConnection</td>
<td>Propriétés du tSQSConnection standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénarios associés</td>
</tr>
<tr>
<td>3962</td>
<td>tSQSInput</td>
<td>Propriétés du tSQSInput standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Récupérer des messages d’une file Amazon SQS</td>
</tr>
<tr>
<td>3971</td>
<td>tSQSMessageChangeVisibility</td>
<td>Propriétés du tSQSMessageChangeVisibility Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénario associé</td>
</tr>
<tr>
<td>3974</td>
<td>tSQSMessageDelete</td>
<td>Propriétés du tSQSMessageDelete Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénario associé</td>
</tr>
<tr>
<td>3977</td>
<td>tSQSOutput</td>
<td>Propriétés du tSQSOutput standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Livrer des messages dans une file Amazon SQS</td>
</tr>
<tr>
<td>3986</td>
<td>tSQSQueueAttributes</td>
<td>Propriétés du tSQSQueueAttributes Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénario associé</td>
</tr>
<tr>
<td>3989</td>
<td>tSQSQueueCreate</td>
<td>Propriétés du tSQSQueueCreate standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénario associé</td>
</tr>
<tr>
<td>3992</td>
<td>tSQSQueueDelete</td>
<td>Propriétés du tSQSQueueDelete standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénario associé</td>
</tr>
<tr>
<td>3995</td>
<td>tSQLTemplateMerge</td>
<td>Propriétés du tSQLTemplateMerge Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénario : Effectuer des opérations de MERGE directement dans le SGBD</td>
</tr>
<tr>
<td>3996</td>
<td>tSQLTemplateMerge</td>
<td>Propriétés du tSQLTemplateMerge Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénario associé</td>
</tr>
<tr>
<td>3999</td>
<td>tSQLTemplateFilterRows</td>
<td>Propriétés du tSQLTemplateFilterRows Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scénario associé</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------------------</td>
<td>---------------------------------</td>
</tr>
</tbody>
</table>
tTeradataInput
Propriétés du tTeradataInput Standard ................................................................. 4113
Scénario associé ........................................................................................................ 4117

tTeradataMultiLoad
Propriétés du tTeradataMultiLoad Standard ............................................................ 4118
Scénario associé ........................................................................................................ 4121

tTeradataOutput
Propriétés du tTeradataOutput Standard ................................................................. 4122
Scénario associé ........................................................................................................ 4128

tTeradataRollback
Propriétés du tTeradataRollback Standard .............................................................. 4129
Scénario associé ........................................................................................................ 4130

tTeradataRow
Propriétés du tTeradataRow Standard ..................................................................... 4131
Scénario associé ........................................................................................................ 4135

tTeradataSCD
Propriétés du tTeradataSCD Standard ..................................................................... 4137
Scénario associé ........................................................................................................ 4140

tTeradataSCDELT
Propriétés du tTeradataSCDELT Standard ............................................................... 4141
Scénario associé ........................................................................................................ 4146

tTeradataTPTExec
Propriétés du tTeradataTPTExec Standard ............................................................. 4147
Attributs facultatifs supportés pour chaque opérateur de consommateur .......... 4151
Scénario : Charger des données dans une base de données Teradata ................. 4152

tTeradataTPTUtility
Propriétés du tTeradataTPTUtility Standard .......................................................... 4160
Scénario associé ........................................................................................................ 4165

tTeradataTPump
Propriétés du tTeradataTPump Standard ............................................................... 4166
Scénario : Insérer des données dans une table d'une base de données Teradata .. 4169

tUniqRow
Propriétés du tUniqRow Standard .......................................................................... 4173
Scénario 1 : Dédoublonner des données ................................................................. 4175

tUnite
Propriétés du tUnite Standard .................................................................................. 4178
Scénario : Itération sur des fichiers et fusion de contenu ...................................... 4179

tVectorWiseCommit
Propriétés du tVectorWiseCommit Standard ........................................................... 4183
Scénario associé ........................................................................................................ 4184

tVectorWiseConnection
Propriétés du tVectorWiseConnection Standard .................................................... 4185
Scénario associé ........................................................................................................ 4187

tVectorWiseInput
Propriétés du tVectorWiseInput Standard ............................................................... 4188
Scénario associé ................................................................................................................................................. 4254

**tVtigerCRMOutput** ........................................................................................................................................... 4255
Propriétés du tVtigerCRMOutput Standard ........................................................................................................ 4255
Scénario associé ................................................................................................................................................. 4257

**tWaitForFile** .................................................................................................................................................... 4258
Propriétés du tWaitForFile Standard .................................................................................................................. 4258
Scénario 1 : Attendre la création d’un fichier et arrêter l’itération après le déclenchement d’un message ................................................................................................................................. 4260
Scénario 2 : Attendre la création d’un fichier et poursuivre l’itération après le déclenchement d’un message ................................................................................................................................................. 4262

**tWaitForSocket** .................................................................................................................................................. 4264
Propriétés du tWaitForSocket Standard ............................................................................................................... 4264
Scénario associé .................................................................................................................................................. 4265

**tWaitForSqlData** ............................................................................................................................................ 4266
Propriétés du tWaitForSqlData Standard ........................................................................................................... 4266
Scénario : Attendre l’ajout de lignes dans une table ............................................................................................ 4268

**tWarn** .............................................................................................................................................................. 4271
Propriétés du tWarn Standard ............................................................................................................................. 4271
Scénarios associés .............................................................................................................................................. 4272

**tWebService** ................................................................................................................................................... 4273
Propriétés du tWebService Standard .................................................................................................................. 4273
Scénario : Obtenir des noms de pays à l’aide d’un service Web ......................................................................... 4276

**tWebServiceInput** .......................................................................................................................................... 4282
Propriétés du tWebServiceInput Standard ........................................................................................................ 4282
Scénario : Obtenir des noms de pays à l’aide d’un Webservice .......................................................................... 4284

**tWriteJSONField** .......................................................................................................................................... 4287
Configurer une arborescence JSON ................................................................................................................... 4287
Propriétés du tWriteJSONField Standard ........................................................................................................ 4288
Scénario : Ecrire des données plates dans des champs JSON ........................................................................... 4289
Scénarios associés .............................................................................................................................................. 4293

**tWriteXMLField** .......................................................................................................................................... 4294
Propriétés du tWriteXMLField Standard ........................................................................................................ 4294
Scénario : Extraire la structure d’un fichier XML et l’insérer dans les champs d’une base de données.............. 4296

**tXMLMap** ....................................................................................................................................................... 4301
Propriétés du tXMLMap Standard ...................................................................................................................... 4301
Mapper et transformer des données de source XML .......................................................................................... 4302
Créer un flux Lookup pour effectuer une jointure sur des données complémentaires .................................. 4309
Mapper des données à l’aide d’un filtre ................................................................................................................ 4315
Capturer les données rejetées par le flux Lookup et le filtre ............................................................................ 4316
Mapper des données à l’aide d’un élément “group” ......................................................................................... 4320
Clasrer les données de sortie avec l’élément “aggregate” ............................................................................... 4323
Scénario 7 : Restructurer des données produit à l’aide de différents éléments de boucle ......................... 4325

**tXMLRPCInput** ............................................................................................................................................ 4335
Propriétés du tXMLRPCInput Standard .......................................................................................................... 4335
Scénario : Chercher le nom d’un Etat via une méthode XMLRPC .................................................................... 4336

**tXSDValidator** ............................................................................................................................................ 4339
Convient à la version 7.0.1. Annule et remplace toute version antérieure de ce guide.

Date de publication : 13 avril 2018

Cette documentation est mise à disposition selon les termes du Contrat Public Creative Commons (CPCC).

Pour plus d’informations concernant votre utilisation de cette documentation en accord avec le Contrat CPCC, consultez : http://creativecommons.org/licenses/by-nc-sa/2.0/.

Mentions légales
Talend est une marque déposée de Talend, Inc.

Tous les noms de marques, de produits, les noms de sociétés, les marques de commerce et de service sont la propriété de leurs détenteurs respectifs.

Licence applicable

tAccessBulkExec

Ce composant permet un gain de performance pendant les opérations d'Insert dans une base de données Access.

Les composants tAccessOutputBulk et tAccessBulkExec sont généralement utilisés ensemble pour écrire des données dans un fichier délimité puis effectuer différentes actions sur le fichier dans une base de données Access, processus en deux étapes. Ces deux étapes sont regroupées dans le composant tAccessOutputBulkExec, détaillé dans une section séparée. L'intérêt d'utiliser un processus en deux étapes réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Le composant tAccessBulkExec effectue une action d'Insert sur les données fournies.

Propriétés du tAccessBulkExec Standard

Ces propriétés sont utilisées pour configurer le tAccessBulkExec s'exécutant dans le framework de Jobs Standard.

Le composant tAccessBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :
Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th><strong>DB Version</strong></th>
<th>Sélectionnez la version d’Access que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| **Username and Password** | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
**None** : n’effectuer aucune opération de table.  
**Drop and create table** : supprimer la table puis en créer une nouvelle.  
**Create table** : créer une table qui n’existe pas encore.  
**Create table if not exists** : créer la table si nécessaire.  
**Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.  
**Clear table** : supprimer le contenu de la table. |
| **Table** | Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit exister pour que l’opération **Insert** soit autorisée. |
| **Local filename** | Nom du fichier à traiter.  
Voir également le *Guide utilisateur de Studio Talend*. |
| **Action on data** | Vous pouvez effectuer, sur les données de la table définie, l’opération que vous souhaitez :  
**Insert** : ajoute de nouvelles entrées à la table. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs. |
### Advanced settings

<table>
<thead>
<tr>
<th><strong>Additional JDBC parameters</strong></th>
<th>Spécifiez des informations supplémentaires de connexion à la base de données existante. Cette option est disponible lorsque la case <strong>Use an existing connection</strong> est décochée dans les <strong>Basic settings</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Include header</strong></td>
<td>Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th><strong>Règle d’utilisation</strong></th>
<th>Ce composant est généralement utilisé avec le composant <strong>tAccessOutputBulk</strong>. Ensemble, ils offrent un gain de performance important pour l’alimentation d’une base de données Access.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton <strong>[+]</strong> pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple...</td>
</tr>
</tbody>
</table>
La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Limitation**

Si vous utilisez un pilote ODBC, assurez-vous que vos versions de JVM et ODBC sont compatibles : 64-bits ou 32-bits pour les deux.

### Scénarios associés

Pour plus d’informations relatives au fonctionnement du composant tAccess BulkExec, consultez les scénarios suivants :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du composant tMysqlOutputBulk.
- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du composant tMysqlOutputBulkExec.
tAccessClose

Ce composant ferme une connexion active à une base de données Access afin de libérer des ressources occupées.

**Propriétés du tAccessClose Standard**

Ces propriétés sont utilisées pour configurer le tAccessClose s’exécutant dans le framework de Jobs Standard.

Le composant tAccessClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant <strong>tAccessConnection</strong> dans la liste.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStat Catcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau des composants.</td>
</tr>
</tbody>
</table>

**Utilisation**

**Règle d’utilisation**

Ce composant est généralement utilisé avec des composants Access, notamment avec le **tAccessConnection** et le **tAccessCommit**.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple...
ple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641** et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520**. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

| Limitation                  | Si vous utilisez un pilote ODBC, vous devez utiliser Java 7 et vous assurer que vos versions de JVM et ODBC sont compatibles : 64-bits ou 32-bits pour les deux. |

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tAccessCommit

Ce composant commite en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes, et permet un gain de performance, en utilisant une connexion unique.

Le composant tAccessCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tAccessCommit Standard

Ces propriétés sont utilisées pour configurer le tAccessCommit s’exécutant dans le framework de Jobs Standard.

Le composant tAccessCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tAccessConnection dans la liste.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. (Avertissement : Si vous utilisez un lien de type Row &gt; Main pour relier le tAccessCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.)</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics   | Cochez cette case pour collecter les données de log au niveau du composant. |
## Utilisation

### Règle d'utilisation

Ce composant est généralement utilisé avec des composants Access et notamment avec les composants `tAccessConnection` et `tAccessRollback`.

### Dynamic settings

Cliquez sur le bouton `[+]` pour ajouter une ligne à la table. Dans le champ `Code`, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste `Component List` de la vue `Basic settings` devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Limitation

Si vous utilisez un pilote ODBC, vous devez utiliser Java 7 et vous assurer que vos versions de JVM et ODBC sont compatibles : 64-bits ou 32-bits pour les deux.

## Scénario associé

Pour plus d’informations relatives au fonctionnement du composant `tAccessCommit`, consultez **Scénario : Insérer des données dans des tables mère/fille** à la page 2620
tAccessConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Le tAccessConnection ouvre une connexion vers une base de données afin d’effectuer une transaction.

Propriétés du tAccessConnection Standard

Ces propriétés sont utilisées pour configurer le tAccessConnection s’exécutant dans le framework de Jobs Standard.

Le composant tAccessConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Use or register a shared DB Connection</td>
<td>Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ Shared DB Connection Name qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela</td>
</tr>
</tbody>
</table>
Vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un composant **tRunJob** ayant une de ces options activée fera échouer votre Job.

### Advanced settings

<table>
<thead>
<tr>
<th>Additional JDBC parameters</th>
<th>Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants <strong>Access</strong> et notamment avec les composants <strong>tAccessCommit</strong> et <strong>tAccessRollback</strong>.</th>
</tr>
</thead>
</table>
| Limitation          | Si vous utilisez un pilote ODBC, vous devez utiliser Java 7 et vous assurer que vos versions de JVM et ODBC sont compatibles : 64-bits ou 32-bits pour les deux.  
Si vous utilisez Java 8, ce composant supporte uniquement le mode **General collation** d’Access. |

### Scénario : Insérer des données dans des tables parent/enfant

Le Job suivant est dédié à des utilisateurs avancés de base de données qui souhaitent effectuer des insertions dans des tables multiples en utilisant une table parent **Table1** pour générer deux tables enfant : **Name** et **Birthday**.

- Créée une base de données connectée à Access 2007 que vous appellerez **Database1**.
- Une fois la base de données Access créée, créez une table appelée **Table1** avec les deux en-têtes de colonne suivants : **Name** et **Birthday**.

Dans la perspective **Integration** de votre **Studio Talend**, le Job est constitué de douze composants, parmi lesquels des composants **tAccessConnection**, **tAccessCommit**, **tAccessInput**, **tAccessOutput** et **tAccessClose**.
• A partir de la Palette, cliquez et déposez les composants suivants dans l’espace de modélisation graphique : un tFileList, un tFileInputDelimited, un tMap, deux tAccessOutput, deux tAccessInput, un tAccessCommit, un tAccessClose et deux tLogRow.

• Connectez le composant tFileList au composant d’entrée tFileInputDelimited à l’aide d’une connexion de type Iterate. Ainsi, le nom du fichier à traiter sera automatiquement renseigné à partir de la liste de fichiers du tFileList en utilisant une variable globale.

• Connectez le composant tFileInputDelimited au tMap et séparez le flux vers les deux composants de sortie tAccessOutput. Utilisez des connexions de type Row pour chacune de ces connexions représentant un flux de données principal.

• Définissez les propriétés du composant tFileList, notamment le répertoire dans lequel sont stockés les fichiers.

• Ajoutez le composant tAccessConnection et connectez-le au composant de départ de ce Job (dans cet exemple, le composant tFileList), à l’aide d’une connexion OnComponentOk afin de définir l’ordre d’exécution.

• Dans la vue Component du composant tAccessConnection, définissez manuellement les informations de connexion ou récupérez-les dans le Repository si vous les avez préalablement stockées dans le dossier DB connection du répertoire Metadata. Pour plus d’informations concernant les Métadonnées, consultez le Guide utilisateur du Studio Talend.

• Dans l’onglet Basic settings du composant tFileInputDelimited, appuyez sur les touches Ctrl + Espace pour accéder à la liste des variables. Définissez le champ File Name avec la variable globale : tFileList_1.CURRENT_FILEPATH. Pour plus d’informations concernant l’utilisation des variables, consultez le Guide utilisateur du Studio Talend.
- Paramétrez les autres champs comme vous le souhaitez, définissez les séparateurs de lignes et de champs (Row et Field Separator) en fonction de la structure de votre document.

- Puis définissez manuellement le schéma du fichier à l'aide du bouton Edit schema ou sélectionnez le schéma dans le Repository. Assurez-vous que le type de données est correctement défini, conformément à la nature des données traitées.

- Dans la zone Output du tMap, ajoutez deux tables de sortie, une que vous nommerez Name pour la table Name, et une deuxième que vous nommerez Birthday, pour la table Birthday. Pour plus d’informations concernant le composant tMap, consultez le Guide utilisateur du Studio Talend.

- Cliquez sur la colonne Name de la zone Input, et glissez-la dans la table Name.

- Cliquez sur la colonne Birthday de la zone Input, et glissez-la dans la table Birthday.

- Puis sélectionnez une connexion de sortie de type Row pour acheminer correctement le flux vers les composants Database de sortie correspondants.

- Dans l’onglet Basic settings des deux composants tAccessOutput, cochez la case Use an existing connection pour récupérer les informations du composant tAccessConnection.

**Remarque :**

Ignorez le champ Commit every puisque cette commande sera annulée par le composant tAccessCommit.
• Nommez votre table dans le champ **Table** en vous assurant que vous saisissez la bonne table. Dans cet exemple, la table est soit *Name*, soit *Birthday*.

• Ne définissez aucune action dans le champ **Action on table** puisque les tables ont déjà été créées.

• Sélectionnez **Insert** dans le champ **Action on data** pour les deux composants de sortie (tAccessOutput).

• Cliquez sur le bouton **Sync columns** pour récupérer le schéma défini dans le **tMap**.

• Connectez ensuite le premier composant tAccessOutput au premier composant tAccessInput à l'aide d’une connexion **OnComponentOk**.

• Dans l’onglet **Basic settings** des deux composants tAccessInput, cochez la case **Use an existing connection** pour récupérer le flux de données distribué. Puis définissez manuellement le schéma du fichier à l’aide du bouton **Edit schema**.

• Nommez ensuite votre table dans le champ **Table Name**. Le nom du tAccessInput_1 sera *Name*.

• Cliquez sur **Guess Query**.

• Connectez les deux composants tAccessInput au composant tLogRow à l’aide d’une connexion de type **Row > Main**. Dans l’onglet **Basic settings** des deux composants tLogRow, sélectionnez **Table** dans le champ **Mode**.

• Ajoutez le composant tAccessConnection sous le composant tFileList et connectez-les à l’aide d’une connexion **OnComponentOk** afin d’arrêter le Job avec les commits de transaction.

• A partir de l’onglet **Basic settings** du composant tAccessCommit, sélectionnez dans la liste déroulante **Component list** la connexion à utiliser, tAccessConnection_1 dans ce scénario.

• Enregistrez votre Job et appuyez sur **F6** pour l’exécuter.
La table parent *Table1* est utilisée à la fois pour générer la table *Name* et la table *Birthday*. 
tAccessInput

Ce composant lit une base de données et en extrait des champs à l’aide de requêtes.

Le tAccessInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux Row > Main.

Propriétés du tAccessInput Standard

Ces propriétés sont utilisées pour configurer le tAccessInput s’exécutant dans le framework de Jobs Standard.

Le composant tAccessInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez la section expliquant comment configurer une connexion à une base de données dans le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

**DB Version**
Sélectionnez la version d’Access que vous utilisez.

**Database**
Nom de la base de données.

**Username et Password**
Informations d’authentification de l’utilisateur de base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Schema et Edit schema**
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.


Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications.
modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Query type et Query**

Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

**Advanced settings**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Additional JDBC parameters</strong></td>
<td>Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
<tr>
<td><strong>Trim all the String/Char columns</strong></td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td><strong>Trim column</strong></td>
<td>Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>QUERY</strong></td>
<td>Requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

**Règle d’utilisation**

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL.
### Dynamic settings

Cliquez sur le bouton `+` pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez [Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte](#) à la page 2641 et [Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement](#) à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le [Guide utilisateur du Studio Talend](#).

### Limitation

Si vous utilisez un pilote ODBC, vous devez utiliser Java 7 et vous assurer que vos versions de JVM et ODBC sont compatibles : 64-bits ou 32-bits pour les deux.

Si vous utilisez Java 8, ce composant supporte uniquement le mode **General collation** d’Access.

---

**Scénarios associés**

Pour un scénario associé, consultez :

Pour plus d’informations sur les contextes dynamiques dans le cadre d’une utilisation de base de données, consultez [tContextLoad](#) à la page 518.
tAccessOutput

Ce composant écrit, met à jour, modifie ou supprime les données d’une base de données.
Le tAccessOutput exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Propriétés du tAccessOutput Standard

Ces propriétés sont utilisées pour configurer le tAccessOutput s’exécutant dans le framework de Jobs Standard.
Le composant tAccessOutput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Remarque</td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par...</td>
</tr>
</tbody>
</table>
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Sélectionnez la version d’Access que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Table      | Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois. |
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
**None** : n’effectuer aucune opération de table.  
**Drop and create table** : supprimer la table puis en créer une nouvelle.  
**Create table** : créer une table qui n’existe pas encore.  
**Create table if not exists** : créer la table si nécessaire.  
**Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.  
**Clear table** : supprimer le contenu de la table. |
| Action on data | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :  
**Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.  
**Update** : Mettre à jour les entrées existantes.  
**Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.  
**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré. |
**Delete** : Supprimer les entrées correspondantes au flux d’entrée.

⚠️ **Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les
modifications à tous les jobs. Si vous souhaitez propager les modifications uniquement au job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

| **Die on error** | Cette case est cochée par défaut et stoppe le job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**. |

### Advanced settings

| **Additional JDBC parameters** | Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.  

**Remarque :** Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies. |

| **Commit every** | Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution. |

| **Additional Columns** | Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.  

**Name** : Saisissez le nom de la colonne à modifier ou à insérer.  

**SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.  

**Position** : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.  

**Reference column** : Saisissez une colonne de référence que le composant **tAccessOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier. |

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

| **Use field options** | Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données. |
Enable debug mode | Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.

Support null in “SQL WHERE” statement | Cochez cette case pour prendre en compte les valeurs Null d’une table de base de données.

**Remarque :**
Assurez-vous que la case Nullable est bien cochée pour les colonnes du schéma correspondantes.

### Global Variables

| Global Variables | NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier. |
| Global Variables | NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier. |
| Global Variables | NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier. |
| Global Variables | NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier. |
| Global Variables | NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier. |
| Global Variables | ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. 

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données Access. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.
**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Limitation**

Si vous utilisez un pilote ODBC, vous devez utiliser Java 7 et vous assurer que vos versions de JVM et ODBC sont compatibles : 64-bits ou 32-bits pour les deux.

Si vous utilisez Java 8, ce composant supporte uniquement le mode **General collation** d’Access.

---

**Scénarios associés**

Pour un scénario associé, consultez :

- **Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput** à la page 2667 du *tMySQLOutput*.
tAccessOutputBulk

Ce composant prépare le fichier à utiliser pour alimenter une base de données Access.

Les composants tAccessOutputBulk et tAccessBulkExec sont généralement utilisés ensemble pour écrire des données dans un fichier délimité puis effectuer différentes actions sur le fichier dans une base de données Access, processus en deux étapes. Ces deux étapes sont regroupées dans le composant tAccessOutputBulkExec, détaillé dans une section séparée. L’intérêt d’utiliser un processus en deux étapes réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Le composant tAccessOutputBulk écrit un fichier délimité.

Propriétés du tAccessOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tAccessOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tAccessOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>File Name</td>
<td>Chemin d’accès et nom du fichier à traiter, et/ou variable à utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Create directory if not exists</td>
<td>Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s’il n’existe pas déjà.</td>
</tr>
</tbody>
</table>
### Append

Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets.

Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Advanced settings

#### Include header

Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.

#### Encoding

Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

#### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.
## Global Variables

| Global Variables | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

## Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec le composant **tAccessBulkExec**. Ensemble, ils offrent un gain de performance important pour l’alimentation d’une base de données Access. |
| Limitation | Si vous utilisez un pilote ODBC, vous devez utiliser Java 7 et vous assurer que vos versions de JVM et ODBC sont compatibles : 64-bits ou 32-bits pour les deux.  
Si vous utilisez Java 8, ce composant supporte uniquement le mode **General collation** d’Access. |

## Scénarios associés

Pour un scénario associé au **tAccessOutputBulk**, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du composant **tMysqlOutputBulk**.
- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du composant **tMysqlOutputBulkExec**.
tAccessOutputBulkExec

Ce composant effectue une action d’Insert sur les données fournies, dans une base de données Access.

Les composants tAccessOutputBulk et tAccessBulkExec sont généralement utilisés ensemble afin d’écrire des données dans un fichier délimité, puis d’effectuer différentes actions sur le fichier dans une base de données Access, processus en deux étapes. Ces deux étapes sont regroupées dans le tAccessOutputBulkExec.

Le tAccessOutputBulkExec est un composant dédié qui permet un gain de performance pendant les opérations d’Insert dans une base de données Access.

Propriétés du tAccessOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tAccessOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tAccessOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Remarque :</td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td>1.</td>
<td>Au niveau parent, enregistrer la connexion à la base de données à partager, dans</td>
</tr>
</tbody>
</table>
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Sélectionnez la version d’Access que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Name</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :&lt;br&gt;&lt;br&gt;<strong>None</strong> : n’effectuer aucune opération de table.&lt;br&gt;<strong>Drop and create table</strong> : supprimer la table puis en créer une nouvelle.&lt;br&gt;<strong>Create table</strong> : créer une table qui n’existe pas encore.&lt;br&gt;<strong>Create table if not exists</strong> : créer la table si nécessaire.&lt;br&gt;<strong>Drop table if exists and create</strong> : supprimer la table si elle existe déjà, puis en créer une nouvelle.&lt;br&gt;<strong>Clear table</strong> : supprimer le contenu de la table.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit déjà exister pour que l’opération d’insert soit autorisée.</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom du fichier à traiter.&lt;br&gt;Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
| Action on data | Vous pouvez effectuer, sur les données de la table définie, l’opération que vous souhaitez :
<br><br>**Insert** : ajoute de nouvelles entrées à la table. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository** |
| **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Create directory if not exists

Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.

Append

Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.

**Advanced settings**

### Additional JDBC parameters

Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.

**Remarque :**

Vous pouvez appuyer sur Ctrl+Espace afin d'accéder à une liste de variables globales prédéfinies.

### Include header

Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.

### Encoding

Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

**Utilisation**

**Règle d’utilisation**

Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans la base de données.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Limitation**

Si vous utilisez un pilote ODBC, assurez-vous que vos versions de JVM et ODBC sont compatibles : 64-bits ou 32-bits pour les deux.

**Scénarios associés**

Pour un scénario associé au tAccessOutputBulkExec, consultez :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du composant tMysqlOutputBulk.
- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec.
tAccessRollback

Ce composant annule la transaction dans une base de données connectée et évite le commit de transaction involontaire.

Propriétés du tAccessRollback Standard

Ces propriétés sont utilisées pour configurer le tAccessRollback s'exécutant dans le framework de Jobs Standard.

Le composant tAccessRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant de connexion tAccessConnection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décrochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en association avec des composants Access, notamment avec le tAccessConnection et le tAccessCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes</td>
</tr>
</tbody>
</table>
bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Limitation

Si vous utilisez un pilote ODBC, vous devez utiliser Java 7 et vous assurer que vos versions de JVM et ODBC sont compatibles : 64-bits ou 32-bits pour les deux.

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tAccessRow

Ce composant exécute des requêtes SQL déclarées sur la base de données spécifiée.

Selon la nature de la requête et de la base de données, tAccessRow agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes. tAccessRow est le composant spécifique à ce type de base de données. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tAccessRow Standard

Ces propriétés sont utilisées pour configurer le tAccessRow s’exécutant dans le framework de Jobs Standard.

Le composant tAccessRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Sélectionnez la version d’Access que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Le schéma est créé et conservé pour ce composant seulement. Voir également le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <strong>No</strong> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td>Table Name</td>
<td>Nom de la table à traiter.</td>
</tr>
</tbody>
</table>
### Query type

La requête peut être **Built-in** ou distante dans le **Repository**.

- **Built-in** : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.
- **Repository** : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.

### Query

Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.

### Die on error

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

#### Propagate QUERY’s recordset

Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.

#### Use PreparedStatement

Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des “?” dans l’instruction SQL définie dans le champ **Query** de l’onglet **Basic settings**.

- **Parameter Index** : Saisissez la position du paramètre dans l’instruction SQL.
- **Parameter Type** : Saisissez le type du paramètre.
- **Parameter Value** : Saisissez la valeur du paramètre.

**Remarque** :
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

#### Commit every

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

#### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.
**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.</td>
</tr>
<tr>
<td>Limitation</td>
<td>Si vous utilisez un pilote ODBC, vous devez utiliser Java 7 et vous assurer que vos versions de JVM et ODBC sont compatibles : 64-bits ou 32-bits pour les deux. Si vous utilisez Java 8, ce composant supporte uniquement le mode General collation d'Access.</td>
</tr>
</tbody>
</table>

**Scénarios associés**

Pour un scénario associé, consultez :

- Procédure du `tDBSQLRow`.
- Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du `tMySQLRow`. 
tAddCRCRow

Ce composant fournit un ID unique afin d’améliorer la qualité des données traitées.

Le tAddCRCRow calcule une clé de substitution basée sur une ou plusieurs colonne(s) et l’ajoute au schéma défini.

**Propriétés du tAddCRCRow Standard**

Ces propriétés sont utilisées pour configurer le tAddCRCRow s’exécutant dans le framework de Jobs Standard.

Le composant tAddCRCRow Standard appartient à la famille Data Quality.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Implication**

Cochez la case correspondant aux colonnes à définir comme clé de substitution et vérifier l’intégrité de leurs données.

**Advanced settings**

<table>
<thead>
<tr>
<th>CRC type</th>
<th>Sélectionnez la longueur CRC. Plus le CRC est long, moins il y aura de risque d’erreur.</th>
</tr>
</thead>
</table>

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>NB_LINE</strong> : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est une étape intermédiaire. Il requiert un flux entrant et une sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

### Scénario : Ajouter une clé de substitution à un fichier

Ce scénario décrit un Job ajoutant une clé de substitution à un schéma de fichier délimité.

![Diagramme de flux](image)

**Construire le Job**

**Procédure**

1. Glissez les composants suivants dans l’espace de modélisation : **tFileInputDelimited**, **tAddCRCRow** et **tLogRow**.
2. Connectez-les en utilisant un lien de type **Main row**.
Configurer le composant d'entrée

Procédure

1. Dans la vue **Component** du tFileInputDelimited, renseignez le champ **File Name** en saisissant le chemin d'accès à votre fichier, ainsi que toutes les propriétés en rapport, au cas où elles ne soient pas stockées dans le **Repository**.

2. Créez le schéma en cliquant sur le bouton **Edit Schema**, si le schéma n'est pas encore stocké dans le **Repository**. Vérifiez la colonne du type de données, et pour remplir le modèle de date (Date pattern), visitez [http://docs.oracle.com/javase/6/docs/api/index.html](http://docs.oracle.com/javase/6/docs/api/index.html) (en anglais).

Configurer le composant tAddCRCRow

Procédure

1. Dans la vue **Component** du composant tAddCRCRow, sélectionnez les cases de la colonne du flux entrant à utiliser pour calculer le CRC.

2. Notez qu'une colonne CRC (en lecture seule) a été ajoutée à la fin du schéma.

3. Dans l'onglet **Advanced settings**, sélectionnez **CRC32** comme **CRC Type** pour avoir une clé de substitution plus longue.
4. Dans la vue **Basic settings** du **tLogRow**, sélectionnez l’option **Table (print values in cells of a table)** pour afficher les données de sortie sous forme de tableau dans la Console.

**Exécuter le Job**

Sauvegardez votre Job et appuyer sur **F6** pour l’exécuter.

<table>
<thead>
<tr>
<th></th>
<th>ID_Owners</th>
<th>Reg_Car</th>
<th>Make</th>
<th>Color</th>
<th>ID_Reseller</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1301</td>
<td>DO 05</td>
<td>Citroen</td>
<td>gold</td>
<td>38</td>
<td>27510715125</td>
</tr>
<tr>
<td>2</td>
<td>2300</td>
<td>ZP 14</td>
<td>Citroen</td>
<td>blue</td>
<td>16</td>
<td>33211434545</td>
</tr>
<tr>
<td>3</td>
<td>4122</td>
<td>JI 74</td>
<td>Renault</td>
<td>yellow</td>
<td>36</td>
<td>11525213515</td>
</tr>
<tr>
<td>4</td>
<td>3395</td>
<td>QP 05</td>
<td>Citroen</td>
<td>yellow</td>
<td>51</td>
<td>14306204562</td>
</tr>
<tr>
<td>5</td>
<td>0029</td>
<td>OF 61</td>
<td>Toyota</td>
<td>red</td>
<td>37</td>
<td>10711350076</td>
</tr>
<tr>
<td>6</td>
<td>4287</td>
<td>VU 44</td>
<td>Citroen</td>
<td>blue</td>
<td>43</td>
<td>25561510712</td>
</tr>
<tr>
<td>7</td>
<td>7119</td>
<td>CQ 97</td>
<td>Honda</td>
<td>yellow</td>
<td>65</td>
<td>1036571035</td>
</tr>
<tr>
<td>8</td>
<td>3761</td>
<td>PA 47</td>
<td>Renault</td>
<td>orange</td>
<td>30</td>
<td>31732350384</td>
</tr>
<tr>
<td>9</td>
<td>9939</td>
<td>CJ 80</td>
<td>Mercedes</td>
<td>red</td>
<td>41</td>
<td>27455544441</td>
</tr>
<tr>
<td>10</td>
<td>7476</td>
<td>KY 03</td>
<td>Citroen</td>
<td>grey</td>
<td>34</td>
<td>27775721061</td>
</tr>
<tr>
<td>11</td>
<td>5287</td>
<td>BP 14</td>
<td>Toyota</td>
<td>green</td>
<td>27</td>
<td>2716661270</td>
</tr>
<tr>
<td>12</td>
<td>0750</td>
<td>OG 65</td>
<td>Toyota</td>
<td>green</td>
<td>8</td>
<td>23636130023</td>
</tr>
<tr>
<td>13</td>
<td>7577</td>
<td>ZQ 59</td>
<td>Volkswagen</td>
<td>purple</td>
<td>55</td>
<td>37277337005</td>
</tr>
</tbody>
</table>

Une colonne CRC supplémentaire a été ajoutée au schéma, calculée d’après toutes les colonnes précédemment sélectionnées (dans ce cas toutes les colonnes du schéma).
tAddLocationFromIP

Ce composant remplace des adresses IP par le lieu géographique correspondant.

Le tAddLocationFromIP localise géographiquement les visiteurs grâce à leurs adresses IP : il identifie le lieu géographique des visiteurs (pays, région, ville, latitude, longitude, code postal, etc.) en utilisant un fichier de consultation d’adresse IP.

**Propriétés du tAddLocationFromIP Standard**

Ces propriétés sont utilisées pour configurer le tAddLocationFromIP s’exécutant dans le framework de Jobs Standard.

Le composant tAddLocationFromIP Standard appartient à la famille Misc.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit les champs à traiter et à passer au</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong> : Le schéma est créé et conservé pour ce composant uniquement. Voir également le Guide utilisateur du Studio Talend .</td>
</tr>
<tr>
<td>Database Filepath</td>
<td>Chemin d’accès au fichier de consultation d’adresses IP.</td>
</tr>
<tr>
<td>Input parameters</td>
<td><strong>Input column</strong> : Sélectionnez la colonne d’entrée à partir de laquelle les valeurs sont collectées pour l’agrégation.</td>
</tr>
<tr>
<td></td>
<td><strong>input value is a hostname</strong> : Cochez cette case si la colonne d’entrée comporte des noms d’hôtes.</td>
</tr>
<tr>
<td></td>
<td><strong>input value is an IP address</strong> : Cochez cette case si la colonne d’entrée comporte des adresses IP.</td>
</tr>
<tr>
<td>Location type</td>
<td><strong>Country code</strong> : Cochez cette case pour remplacer l’adresse IP par le code du pays correspondant.</td>
</tr>
<tr>
<td></td>
<td><strong>Country name</strong> : Cochez cette case pour remplacer l’adresse IP par le nom du pays correspondant.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |
Global Variables


Utilisation

| Règle d’utilisation | Ce composant peut être utilisé comme étape intermédiaire du traitement permettant de remplacer des adresses IP par des informations géolocalisées. Il ne peut être utilisé comme composant de début car il nécessite un flux en entrée, ainsi qu’un composant de sortie. |
  • geoip.jar |

Scénario : Identifier la localisation géographique d’une adresse IP

Le scénario Java suivant crée un Job à trois composants associant une adresse IP à un lieu géographique. Il permet d’obtenir la localisation géographique du visiteur d’un site à partir de son adresse IP.
Déposer et relier les composants

Procédure
1. À partir de la Palette, cliquez-déposez les composants suivants dans l’éditeur graphique : le tFixedFlowInput, le tAddLocationFromIP et le tLogRow.
2. Connectez les trois composants en utilisant des liens de type Row Main.

Configurer les composants

Procédure
1. Dans l’espace de modélisation graphique, sélectionnez le tFixedFlowInput et cliquez sur l’onglet Component pour définir la configuration de base (Basic settings) du tFixedFlowInput.
2. Cliquez sur le bouton [...] à côté du champ Edit Schema pour définir les données que vous voulez utiliser en entrée. Dans ce scénario, le schéma est constitué d’une colonne comportant une adresse IP.
3. Cliquez sur le bouton OK pour fermer la boîte de dialogue, puis acceptez la propagation des modifications lorsque le système vous invite à le faire. La colonne définie s’affiche dans la section Values de la vue Basic settings.
4. Dans le champ Number of rows, saisissez le nombre de lignes à générer puis cliquez dans la cellule Value et définissez la valeur de l’adresse IP.
5. Dans l’espace de modélisation graphique, sélectionnez le composant tAddLocationFromIP et cliquez dans la vue Component pour définir la configuration de base (Basic settings) du tAddLocationFromIp.

6. Cliquez sur le bouton Sync columns pour synchroniser le schéma avec le schéma d’entrée du tFixedFlowInput.

7. Parcourez vos dossiers jusqu’au fichier GeoIP.dat pour en définir le chemin d’accès dans le champ Database filepath.

Remarque :
Assurez-vous de télécharger la dernière version du fichier de consultation des adresses IP, à partir du site indiqué dans la vue Basic settings du composant tAddLocationFromIp.

8. Dans la zone Input parameters, définissez les paramètres d’entrée selon vos besoins. Dans ce scénario, la colonne d’entrée correspond à la colonne ip, qui comporte une adresse IP, définie précédemment.

9. Dans la zone Location type, définissez le type de lieu selon vos besoins. Dans ce scénario, vous allez afficher le nom du pays.

10. Dans l’éditeur graphique, sélectionnez le composant tLogRow. Cliquez sur la vue Component puis définissez la configuration de base (Basic settings) du tLogRow selon vos besoins. Dans ce scénario, vous allez afficher les valeurs dans les cellules d’une table.

Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur les touches Ctrl+S pour sauvegarder votre Job.
2. Appuyez sur F6 ou cliquez sur le bouton Run, dans la vue Run, pour exécuter le Job.

Résultats

La ligne unique générée par le Job affiche le nom du pays correspondant à l’adresse IP renseignée.
tAdvancedFileOutputXML

Ce composant crée un fichier XML contenant les valeurs des données définies dans l’arborescence XML.

Le composant tAdvancedFileOutputXML transmet des données vers un fichier de type XML et propose une interface de gestion des boucles et des Group by si nécessaire.

Propriétés du tAdvancedFileOutputXML Standard

Ces propriétés sont utilisées pour configurer le tAdvancedFileOutputXML s’exécutant dans le framework de Jobs Standard.

Le composant tAdvancedFileOutputXML Standard appartient aux familles File et XML.

Le composant de ce framework est toujours disponible.

Basic settings

| Property type | Peut être Built-in ou Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|              | • View schema : sélectionnez cette option afin de voir le schéma.  
|              | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
|              | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content. |
| Built-in     | Propriétés utilisées ponctuellement. |
| Repository   | Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |
| Use Output Stream | Cochez la case pour traiter le flux de données qui vous intéresse. Une fois cochée, le champ Output Stream s’affiche et vous pouvez saisir le flux de données souhaité.  
|                | Le flux de données à traiter doit être ajouté au flux afin que ce composant récupère ces données via la variable représentative correspondante.  
|                | Cette variable peut être prédéfinie dans votre Studio Talend ou fournie par le contexte ou les composants utilisé(s) avec ce composant. Sinon, vous pouvez la |
définir manuellement et l’utiliser selon votre Job, par exemple à l’aide d’un **Java** ou d’un **JavaFlex**.

Afin d’éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d’autocomplétion (**Ctrl+Espace**) afin de remplir le champ, si cette variable a été correctement définie.

Pour plus d’informations concernant l’utilisation d’un flux, consultez **Scénario 2 : Lire les données d’un fichier distant en mode stream** à la page 1072.

<table>
<thead>
<tr>
<th><strong>File name</strong></th>
<th>Chemin d’accès et nom du fichier de sortie, et/ou variable à utiliser. Ce champ est indisponible si vous avez coché la case <strong>Use Output Stream</strong>. Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Configure XML tree</strong></td>
<td>Ouvre l’interface d’aide à la création du fichier XML. Pour plus d’informations sur l’interface, consultez <strong>Définir un arbre XML</strong> à la page 114.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champ qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. <strong>Built-in</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <strong>Guide utilisateur du Studio Talend</strong>. <strong>Repository</strong> : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
<tr>
<td><strong>Sync columns</strong></td>
<td>Cliquez sur ce bouton pour synchroniser le schéma de sortie avec celui d’entrée. La fonction <strong>Sync</strong> ne s’affiche que si une connexion de type Row est liée au composant de sortie.</td>
</tr>
<tr>
<td><strong>Append the source xml file</strong></td>
<td>Cochez cette option pour ajouter de nouvelles lignes à la fin de votre fichier source XML.</td>
</tr>
<tr>
<td><strong>Generate compact file</strong></td>
<td>Cochez cette case pour générer un fichier ne comprenant aucun espace vide ni aucun séparateur de ligne. L’ensemble des éléments se présente alors sur une ligne unique, ce qui permet de réduire le poids du fichier.</td>
</tr>
<tr>
<td><strong>Include DTD or XSL</strong></td>
<td>Cochez cette case pour ajouter la déclaration DOCTYPE indiquant le nom de l’élément racine et le chemin d’accès et le nom du fichier DTD ou pour ajouter l’instruction de traitement indiquant le type de feuille de style (telle que les XSL) utilisée et son chemin d’accès et nom.</td>
</tr>
</tbody>
</table>
**Advanced settings**

| **Split output in several files** | Si le fichier XML de sortie est lourd, vous pouvez le scinder en plusieurs fichiers en définissant un certain nombre de lignes par fichier. |
| **Trim Data** | Cette case est activée lorsque vous utilisez le mode de génération dom4j. Cochez cette case pour supprimer les espaces en début et en fin de valeur d’un élément XML. |
| **Create directory only if not exists** | Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier XML de sortie s’il n’existe pas déjà. |
| **Create empty element if needed** | Cette case est cochée par défaut. Si le contenu de la colonne **Associated Column** est nul ou si aucune colonne n’est associée au nœud XML, cette option créera une balise ouvrante et une balise fermante aux endroits prévus. |
| **Create attribute even if its value is NULL** | Cochez cette case pour générer l’attribut de la balise XML pour la colonne d’entrée associée dont la valeur est null. |
| **Create attribute even if it is unmapped** | Cochez cette case pour générer l’attribut de la balise pour la colonne d’entrée associée non mappée. |
| **Create associated XSD file** | Si l’un des éléments XML est associé à un espace de nommage, cette option créera le fichier XSD correspondant.  

**Remarque :**  
Pour utiliser cette option, vous devez sélectionner le mode de génération **Dom4J** dans la liste **Generation mode**. |
| **Add Document type as node** | Cochez cette case pour ajouter des colonnes de type **Document** dans le nœud au lieu d’échapper des chaînes de caractère dans le fichier XML de sortie.  
Cette case apparaît uniquement lorsque le mode de génération est configuré à **Slow and memory-consuming (Dom4j)** dans l’onglet **Advanced settings**. |
| **Advanced separator (for number)** | Cochez cette option pour modifier les séparateurs utilisés pour les nombres :  
**Thousands separator** : définissez le séparateur utilisé pour les milliers. |
**Generation mode**

Sélectionnez le mode de génération correspondant à votre mémoire disponible. Les modes disponibles sont :

- **Lent et consommateur de mémoire** *(Slow and memory-consuming - Dom4J)*.
  
  **Remarque :**
  Cette option vous permet d’utiliser Dom4J pour traiter des fichiers XML très complexes.

- **Rapide et peu consommateur de mémoire** *(Fast with low memory consumption)*.

Lorsque la case **Append the source xml file**, dans la vue **Basic settings** est cochée, ce champ disparaît car le mode de génération dom4j est automatiquement sélectionné.

**Encoding**

Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

**Don’t generate empty file**

Cochez cette case pour annuler la génération du fichier si celui-ci est vide.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

---

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
Utilisation

| Règle d'utilisation | Utilisez ce composant pour créer un fichier XML à partir des données transmises par d'autres composants via une connexion de type Row. |

Définir un arbre XML

Double-cliquez sur le composant `tAdvancedFileOutputXML` pour ouvrir l'interface dédiée ou cliquez sur le bouton `[...]` du champ `Configure Xml Tree` de l'onglet `Basic settings` dans la vue `Component`.

Le champ `Schema List` de la zone `Linker Source`, à gauche de l'interface, contient toutes les colonnes du flux d'entrée (à condition que le flux d'entrée soit connecté au composant `tAdvancedFileOutputXML`).

A droite de l'interface, dans la zone `Linker Target`, définissez la structure XML que vous souhaitez obtenir en sortie.

Vous pouvez la créer manuellement ou tout simplement importer la structure XML. Puis importez les colonnes du schéma d'entrée dans l'élément de l'arbre XML correspondant.

Importer un arbre XML

Le meilleur moyen de renseigner l'arbre XML est d'importer un fichier XML bien formé.

Procédure

1. Renommez la balise racine qui s'affiche par défaut dans le panneau `XML tree`, en cliquant sur celle-ci.
2. Dans la colonne `XML Tree`, cliquez-droit sur le champ `root tag` pour afficher le menu contextuel.
3. Dans le menu, sélectionnez **Import XML tree**.
4. Sélectionnez le fichier à importer et cliquez sur **OK**.
   - Vous pouvez importer la structure XML d'un fichier au format XML, XSD et DTD.
   - Lors de l'import d'une structure XML d'un fichier XSD, vous pouvez choisir un élément comme la racine de votre arbre XML.
   La colonne **XML Tree** est donc automatiquement renseignée avec les éléments.
5. Si vous devez ajouter ou supprimer un élément ou un sous-élément, cliquez-droit sur l'élément correspondant, dans l'arborescence, pour afficher le menu contextuel.
6. Sélectionnez **Delete** pour supprimer la sélection de l'arbre ou sélectionnez l'option adéquate parmi les suivantes : **Add sub-element, Add attribute, Add namespace** pour enrichir l'arbre.

**Créer manuellement l'arbre XML**

Si vous ne possédez pas de structure XML déjà définie, vous pouvez la créer manuellement.

**Procédure**

1. Dans la colonne **XML Tree**, cliquez une fois sur le champ **root tag** pour le renommer.
2. Cliquez-droit sur ce champ pour afficher le menu contextuel.
3. Dans le menu, sélectionnez **Add sub-element** pour créer le premier élément de la structure.
4. Si vous devez ajouter un attribut ou un élément fils à un élément, ou si vous devez supprimer un élément, cliquez-droit à gauche du nom de l'élément correspondant, pour afficher le menu contextuel.
5. Dans le menu, sélectionnez l'option adéquate parmi les suivantes : **Add sub-element, Add attribute, Add namespace ou Delete**.

**Mapping de données XML**

Une fois votre arbre XML créé, vous pouvez alimenter chaque élément ou sous-élément XML avec les colonnes du flux d'entrée dans la colonne **Related Column**.

**Procédure**

1. Cliquez sur une des entrées de la colonne **Schema List**.
2. Glissez-la dans le sous-élément correspondant à droite.
3. Relâchez-la pour que le mapping soit effectif.
4. Si vous devez déconnecter n'importe quel mapping de n'importe quel élément de l'arbre XML, sélectionnez l'élément et cliquez-droit à gauche de son nom pour afficher le menu contextuel.
5. Sélectionnez **Disconnect linker**.

**Définir le statut du nœud**

Définir l'arbre XML et le transfert de données ne suffit pas. Vous devez aussi définir l'élément sur lequel la boucle est effectuée et si nécessaire l'élément à partir duquel le regroupement est effectué.
Définir un élément Boucle

L’élément Boucle permet de définir l’élément objet d’une itération. L’élément Boucle est généralement un générateur de lignes.

Pourquoi et quand exécuter cette tâche
Pour définir un élément comme un élément Boucle :

Procédure
1. Sélectionnez l’élément adéquat dans l’arbre XML.
2. Cliquez-droit à gauche du nom de l’élément pour afficher le menu contextuel.
3. Sélectionnez l’option Set as Loop Element.

Résultats
La colonne Node Status affiche le nouveau statut sélectionné.

Remarque :
Il ne peut y avoir qu’un seul élément Boucle à la fois.

Définir un élément Group

L’élément Group est optionnel, il représente un élément constant sur lequel est effectuée une fonction Groupby. L’élément Group ne peut être défini que si un élément Boucle a été préalablement défini.

Pourquoi et quand exécuter cette tâche
Lorsque vous utilisez un élément Group, les lignes doivent être triées pour pouvoir être regroupées par le nœud sélectionné.

Pour définir un élément comme élément Group :

Procédure
1. Sélectionnez l’élément adéquat dans l’arbre XML.
2. Cliquez-droit à gauche du nom de l’élément pour afficher le menu contextuel.
3. Sélectionnez l’option Set as Group Element.

Résultats
La colonne Node Status affiche le nouveau statut sélectionné et tout autre statut de regroupement est automatiquement défini, si nécessaire.

Une fois le mapping terminé, cliquez sur OK pour valider les paramètres et continuez la configuration de votre Job.

Scénario : Créer un fichier XML à l’aide d’une boucle

Le scénario suivant décrit la création d’un fichier XML à partir d’un fichier plat trié contenant une vidéothèque.
Le fichier XML de sortie affiche la structure telle qu'elle a été définie.

**Configurer le fichier source**

**Procédure**

1. Dans la Palette de composants, cliquez sur les composants \texttt{tFileInputDelimited} et \texttt{tAdvancedFileOutputXML} et déposez-les dans l'espace de modélisation.

2. Sinon, si vous avez déjà créé une métadonnée contenant la description du fichier d'entrée dans le nœud \texttt{Metadata} du Repository, alors vous pouvez cliquer-déposer directement cette métadonnée dans l'espace de modélisation pour paramétrer automatiquement le flux d'entrée.

3. Cliquez-droit sur le composant d'entrée et reliez-le au composant \texttt{tAdvancedFileOutputXML} à l'aide d'une connexion de type Row Main.

4. Sélectionnez le composant \texttt{tFileInputDelimited} et paramétrez ses propriétés dans la vue Component en bas de la fenêtre du Studio.

5. Dans la liste déroulante Property Type, sélectionnez l'option Repository si la description de votre fichier est stockée dans une métadonnée du Repository. Si vous cliquez-déposez le composant directement à partir de la métadonnée, vous n'aurez pas besoin de modifier ses propriétés.

   Sinon, sélectionnez l'option Built-In et renseignez manuellement les champs de l'onglet Basic settings.

6. Le fichier d'entrée contient les colonnes suivantes, séparées par des points-virgules : \texttt{id}, \texttt{name}, \texttt{category}, \texttt{year}, \texttt{language}, \texttt{director} et \texttt{cast}. 
Dans cette exemple, le champ **Cast** regroupe différentes valeurs et **id** s'incrémente lorsque l'on change de film.

7. Si nécessaire, définissez le schéma du **tFileDelimitedInput** en fonction de la structure du fichier.

8. Une fois que vous avez vérifié que le schéma du fichier d’entrée correspond à vos attentes, cliquez sur **OK** pour valider.

### Configurer la sortie XML et le mapping

#### Procédure

1. Puis sélectionnez le composant **tAdvancedFileOutputXML** et cliquez sur la vue **Component** pour paramétrer ses propriétés, ainsi que le mapping dans l’onglet **Basic settings**. Notez que vous pouvez directement double-cliquer sur le composant pour ouvrir l’interface de mapping.
2. Dans le champ **File Name**, sélectionnez le fichier qui sera écrit à la fin du Job, s’il existe déjà ou saisissez le chemin d’accès et le nom du fichier qui seront créés automatiquement pour la sortie.
Par défaut, le schéma (description du fichier) est automatiquement propagé à partir du flux d’entrée. Mais vous pouvez le modifier si nécessaire.

3. Puis cliquez sur le bouton [...] ou double-cliquez sur le composant **tAdvancedFileOutputXML** dans l’espace de modélisation pour ouvrir l’éditeur de mapping.
Les colonnes du schéma du fichier d’entrée sont listées à gauche de l’interface.

Vous pouvez soit créer la structure nœud par nœud. Pour plus d’informations sur la création manuelle de l’arbre XML, consultez **Définir un arbre XML** à la page 114.
Dans cet exemple, un modèle XML est utilisé pour renseigner automatiquement l’arbre XML.

5. Cliquez-droit sur le champ **root tag** qui apparaît par défaut et sélectionnez l’option **Import XML tree** à la fin des options du menu contextuel.

6. Dans la fenêtre [Ouvrir], sélectionnez le fichier XML à importer et cliquez sur **OK** pour valider l’import.

**Remarque :**
Vous pouvez importer une arborescence XML à partir de fichiers au format XML, XSD et DTD.

7. Puis cliquez-déposez chaque champ de la colonne **Schema List** vers son élément correspondant dans l’arbre XML de la colonne **XML tree** tel que décrit dans **Mapping de données XML** à la page 115.
Le mapping apparaît sous la forme d’une flèche bleue entre la zone de gauche et celle de droite.
Enfin, dans la colonne **Node Status**, définissez le statut du nœud sur lequel la boucle est effectuée. Dans cet exemple, l’élément *Cast* correspondant à l’élément *Boucle* puisque les données de ce champ dans le fichier d’entrée ne sont jamais les mêmes.

Cliquez-droit sur l’élément *Cast* de l’arbre XML et sélectionnez l’option **Set as loop element**.

8. Pour regrouper les données par film, l’élément *Movie* doit aussi être défini comme élément *Group*.

    Cliquez-droit sur le nœud parent *Movie* de l’arbre XML et sélectionnez l’option **Set as group element**.

    Le statut sélectionné paraît sur la ligne de l’élément correspondant dans la colonne **Node Status**.

9. Cliquez sur **OK** pour valider la configuration.

10. Appuyez sur **F6** pour exécuter le Job.
<?xml version="1.0" encoding="ISO-8859-15"?>

<VideoCollection>
  <Movie id="1">
    <Category>Comedy</Category>
    <Year>1982</Year>
    <Language>English</Language>
    <Title>Footsie</Title>
    <Crew>
      <Director>Sydney Pollack</Director>
      <Cast>Dustin Hoffman</Cast>
      <Cast>Jessica Lange</Cast>
      <Cast>Sydney Pollack</Cast>
    </Crew>
  </Movie>
  <Movie id="2">
    <Category>Science Fiction</Category>
    <Year>1997</Year>
    <Language>French</Language>
    <Title>The 5th Element</Title>
    <Crew>
      <Director>Eric Besson</Director>
      <Cast>Bruce Willis</Cast>
    </Crew>
  </Movie>
</VideoCollection>
tAggregateRow

Ce composant reçoit un flux de données et fait une agrégation basée sur une ou plusieurs colonnes.
Pour chacune des lignes en sortie, une clé d’agrégation est fournie, ainsi que le résultat de l’opération d’agrégation correspondant (min, max, sum, etc.).
Le tAggregateRow permet d’établir des métriques et des statistiques basées sur des valeurs ou des calculs.

Propriétés du tAggregateRow Standard

Ces propriétés sont utilisées pour configurer le tAggregateRow s’exécutant dans le framework de Jobs Standard.
Le composant tAggregateRow Standard appartient à la famille Processing.
Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Repository</td>
<td>Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Group by</td>
<td>Définit les ensembles d’agrégation, dont les valeurs sont utilisées pour les calculs.</td>
</tr>
</tbody>
</table>
**Output Column** : Sélectionnez le libellé de colonne dans la liste fournie, basée sur la structure de schéma que vous avez définie. Vous pouvez ajouter autant de colonnes de sortie que vous le souhaitez afin d’affiner les agrégations.

Ex : Sélectionnez Country (Pays) pour calculer la moyenne des valeurs pour chaque pays ou sélectionnez Country et Region pour comparer les résultats des régions d’un pays par rapport aux régions d’un autre pays.

**Input Column** : Faites la correspondance entre les libellés des colonnes d’entrée avec ceux des colonnes de sortie, dans le cas où vous souhaitez que les libellés du schéma de sortie soient différents du schéma d’entrée.

**Operations** : Sélectionnez le type d’opération à effectuer ainsi que la valeur à utiliser pour le calcul et le champ de sortie.

**Output Column** : Sélectionnez le champ de destination dans la liste.

**Function** : Sélectionnez l’opérateur parmi :
- **count** : calcule le nombre de ligne,
- **min** : sélectionne la plus petite valeur,
- **max** : sélectionne la plus grande valeur,
- **avg** : calcule la moyenne,
- **sum** : calcule la somme,
- **first** : retourne la première valeur,
- **last** : retourne la dernière valeur,
- **list** : liste les valeurs d’une agrégation de plusieurs clés,
- **list (object)** : liste les valeurs Java d’une agrégation de plusieurs clés,
- **count (distinct)** : compte le nombre de lignes sans les doublons,
- **standard deviation** : calcule l’écart-type.
- **union (geometry)** : fusionne un ensemble d’objets Geometry.

**Input column position** : Sélectionnez la colonne d’entrée à partir de laquelle les valeurs sont collectées pour l’agrégation.

**Ignore null values** : Cochez cette case devant le nom de toutes les colonnes pour lesquelles vous voulez que les valeurs nulles soient ignorées.
### Advanced settings

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delimiter (only for list operation)</td>
<td>Saisissez le séparateur que vous souhaitez utiliser afin de séparer les différentes opérations.</td>
</tr>
<tr>
<td>Use financial precision, this is the max precision for &quot;sum&quot; and &quot;avg&quot; operations, checked option heaps more memory and slower than unchecked.</td>
<td>Cochez cette case afin d'utiliser une précision financière, une précision maximale, mais qui consomme plus de mémoire et rend le processus plus lent.</td>
</tr>
</tbody>
</table>
| Use financial precision, this is the max precision for "sum" and "avg" operations, checked option heaps more memory and slower than unchecked. | Avertissement :  
Il est conseillé d'utiliser le type BigDecimal en sortie avec cette option, afin d'obtenir des résultats précis. |
| Check type overflow (slower)                                | Vérifie les types des données afin d'éviter que le Job ne plante.                                                                        |
| Check ULP (Unit in the Last Place), ensure that a value will be incremented or decremented correctly, only float and double types. (slower) | Cochez cette case afin d'avoir la meilleure précision possible pour les types Float et Double.                                          |
| tStatCatcher Statistics                                     | Cochez cette case pour collecter les données de log au niveau du composant. Notez que cette case n’est pas disponible dans la version Map/Reduce de ce composant. |

### Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.  
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------|

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant intermédiaire car il traite un flux de données, par conséquent il requiert des composants en entrée comme en sortie. Généralement, l’utilisation du composant tAggregateRow est combinée au composant tSortRow.</th>
</tr>
</thead>
</table>
Agréger des valeurs et trier des données

Cet exemple vous montre comment utiliser des composants Talend pour agréger les résultats totaux d’étudiants et trier les résultats agrégés selon le nom des étudiants.

Créer un Job pour agréger et trier les données

Créez un Job pour agréger les résultats totaux des étudiants à l’aide du composant `tAggregateRow`, trier les données agrégées à l’aide du `tSortRow`, puis pour afficher les données agrégées et triées dans la console.

**Procédure**

1. Créez un nouveau Job et ajoutez un composant `tFixedFlowInput`, un `tAggregateRow`, un `tSortRow` et un `tLogRow` en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le composant `tFixedFlowInput` au `tAggregateRow` à l’aide d’un lien `Row > Main`.
3. Répétez l’opération pour relier le `tAggregateRow` au `tSortRow` et le `tSortRow` au `tLogRow`.

Configurer le Job pour agréger et trier les données

Configurez le Job pour agréger les résultats totaux des étudiants à l’aide du `tAggregateRow` et trier les données agrégées à l’aide du composant `tSortRow`.

**Procédure**

1. Double-cliquez sur le composant `tFixedFlowInput` pour ouvrir sa vue `Basic settings`.
2. Cliquez sur le bouton 📊 à côté du champ `Edit schema` pour ouvrir la fenêtre de schéma et le définir, en ajoutant deux colonnes, `name`, de type `String` et `score`, de type `Double`. Cela fait, cliquez sur OK pour sauvegarder les modifications et fermer la fenêtre du schéma.
3. Dans la zone Mode, sélectionnez **Use Inline Content (delimited file)** et, dans le champ Content qui s’affiche, saisissez les données d’entrée suivantes :

```
Peter;92
James;93
Thomas;91
Peter;94
James;96
Thomas;95
Peter;96
James;92
Thomas;98
Peter;95
James;96
Thomas;93
Peter;98
James;97
Thomas;95
```
4. Double-cliquez sur le composant **tAggregateRow** pour ouvrir sa vue **Basic settings**.

5. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir la fenêtre du schéma et définissez le schéma en ajoutant cinq colonnes, **name**, de type **String** et **sum**, **average**, **max** et **min**, de type **Double**.

Cela fait, cliquez sur **OK** pour sauvegarder les modifications et fermer la fenêtre du schéma.

6. Ajoutez une ligne à la table **Group by** en cliquant sur le bouton [+ ] sous la table et sélectionnez **name** dans les colonnes **Output column** et **Input column position**, afin de grouper les données d’entrée selon la colonne **name**.

7. Ajoutez quatre lignes à la table **Operations** et définissez les opérations à effectuer. Dans cet exemple, les opérations sont **sum**, **average**, **max** et **min**. Sélectionnez **score** dans les quatre lignes de la colonne **Input column position** afin d’agréger les données d’entrée selon les résultats.

8. Double-cliquez sur le **tSortRow** pour ouvrir sa vue **Basic settings**.
9. Ajoutez une ligne à la table **Criteria** et spécifiez la colonnes sur laquelle baser l’opération de tri. Dans cet exemple, la colonne *name*. Sélectionnez *alpha* dans la colonne *sort num or alpha?*, puis *asc* dans la colonne *Order asc or desc?*, afin de trier les données agrégées par ordre alphabétique ascendante.

10. Double-cliquez sur le composant **tLogRow** pour ouvrir sa vue **Basic settings** et sélectionnez **Table** *(print values in cells of a table)* dans la zone **Mode**, pour une lisibilité optimale des résultats.

**Exécuter le Job pour agréger et trier les données**

Après avoir configuré le Job et ses composants pour agréger et trier les données, vous pouvez exécuter le Job et vérifier ses résultats d'exécution.

**Procédure**

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder le Job.
2. Appuyez sur **F6** pour l’exécuter.

```plaintext
[statistics] connecting to socket on port 3914
[statistics] connected
-----------+-----------------------
tLogRow_1
-----------+-----------------------
name | sum  | average | max  | min  
-----------+-----------------------
James  | 474.0 | 94.8    | 97.0 | 92.0 |
Peter  | 475.0 | 95.0    | 98.0 | 92.0 |
Thomas | 472.0 | 94.4    | 98.0 | 91.0 |
-----------+-----------------------
[statistics] disconnected
```

**Résultats**

Comme affiché ci-dessus, les résultats totaux des étudiants sont agrégés et triés par ordre alphabétique ascendant selon le nom des étudiants.
tAggregateSortedRow

Ce composant reçoit un flux de données triées sur lequel il effectue une agrégation basée sur une ou plusieurs colonnes.

La clé d’agrégation et les résultats pertinents des opérations (min, max, sum, etc.) sont dispensés pour chaque ligne de sortie. Étant donné que les données du flux d’entrée sont supposées être déjà triées, les performances n’en sont que plus optimisées.

Le tAggregateSortedRow vous permet de récupérer un échantillon de métriques reposant sur des valeurs ou des calculs.

Propriétés du tAggregateSortedRow Standard

Ces propriétés sont utilisées pour configurer le tAggregateSortedRow s’exécutant dans le framework de Jobs Standard.

Le composant tAggregateSortedRow Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|                   | • View schema : sélectionnez cette option afin de voir le schéma.
|                   | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
|                   | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. Cliquez sur Sync columns pour récupérer le schéma du composant précédent connecté dans le Job.
| Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers |

| **Input rows count** | Spécifiez le nombre de lignes envoyé au composant tAggregateSortedRow.  

**Remarque :**  
Si vous spécifiez une limite (dans le champ Limit) au nombre de lignes à traiter dans le composant d’entrée, vous devez utiliser cette même limite dans le champ **Input rows count**. |

| **Group by** | Définit les ensembles d’agrégation, dont les valeurs sont utilisées pour les calculs.  

| **Output Column** : Sélectionnez le libellé de colonne dans la liste fournie, basée sur la structure de schéma que vous avez définie. Vous pouvez ajouter autant de colonnes de sortie que vous le souhaitez afin d’affiner les agrégations.  

Exemple : Sélectionnez Country (Pays) pour calculer la moyenne des valeurs pour chaque pays ou sélectionnez Country et Region pour comparer les résultats des régions d’un pays par rapport aux régions d’un autre pays. |

| **Input Column** : Faites la correspondance entre les libellés des colonnes d’entrée avec ceux des colonnes de sortie, dans le cas où vous souhaitez que les libellés du schéma de sortie soient différents du schéma d’entrée. |

| **Operations** | Sélectionnez le type d’opération à effectuer ainsi que la valeur à utiliser pour le calcul et le champ de sortie. |

| **Output Column** : Sélectionnez le champ de destination à partir de la liste déroulante.  

**Function** : Sélectionnez l’opérateur parmi :  
- **count** : calcule le nombre de ligne,  
- **min** : sélectionne la plus petite valeur,  
- **max** : sélectionne la plus grande valeur,  
- **avg** : calcule la moyenne,  
- **sum** : calcule la somme,  
- **first** : retourne la première valeur,  
- **last** : retourne la dernière valeur,  
- **list** : liste les valeurs d’une agrégation de plusieurs clés,  
- **list (object)** : liste les valeurs Java d’une agrégation de plusieurs clés,  
- **count (distinct)** : compte le nombre de lignes sans les doublons,  
- **standard deviation** : calcule l’écart-type. |
• union (geometry) : fusionne un ensemble d'objets Geometry.

Input column position : Sélectionnez la colonne d'entrée à partir de laquelle les valeurs sont récupérées avant d'être agrégées.

Ignore null values : Cochez cette case devant le nom de toutes les colonnes pour lesquelles vous voulez que les valeurs nulles soient ignorées.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant. |

**Global Variables**

| Global Variables | ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
|                 | NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| Règle d'utilisation | Ce composant traite des flux de données, il nécessite donc des composants d'entrée et sortie. Il est donc défini en tant que composant intermédiaire. |

**Scénario : trier et agréger les données d'entrée**

Ce scénario décrit un Job triant les entrées des données d’entrée selon deux colonnes et affiche les données triées dans la console, agrège les données triées à partir d'une colonne et affiche ces données agrégées dans la console.
Ajouter et relier les composants

**Procédure**

1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette dans l’espace de modélisation graphique : un *tFixedFlowInput*, un *tSortRow*, un *tAggregateSortedRow* et deux *tLogRow*.
2. Reliez le *tFixedFlowInput* au *tSortRow* à l’aide d’un lien Row > Main.
3. Répétez l’opération afin de relier le *tSortRow* au premier *tLogRow*, reliez le premier *tLogRow* au *tAggregateSortedRow* et reliez le *tAggregateSortedRow* au second *tLogRow*.

Configurer les composants

**Trier les données d’entrée**

**Procédure**

1. Double-cliquez sur le *tFixedFlowInput* pour ouvrir sa vue *Basic settings*.

2. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans la fenêtre qui s’ouvre, définissez le schéma en ajoutant quatre colonnes : *Id* et *Age* de type *Integer*, ainsi que *Name* et *Team* de type *String*.
Cliquez sur OK pour fermer l’éditeur de schéma. Acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

3. Dans la zone Mode, sélectionnez Use Inline Content(delimited file) et, dans le champ Content affiché, saisissez les données d’entrée à trier et agréger. Dans cet exemple, les données d’entrée se présentent comme suit :

1;Thomas;28;Component Team
2;Harry;32;Doc Team
3;John;26;Component Team
4;Nicolas;27;QA Team
5;George;24;Component Team
6;Peter;30;Doc Team
7;Teddy;23;QA Team
8;James;26;Component Team

4. Double-cliquez sur le tSortRow pour ouvrir sa vue Basic settings.

5. Cliquez sur le bouton [+ ] sous la table Criteria pour ajouter autant de lignes que nécessaire et spécifiez les critères de tri dans la table. Dans cet exemple, deux lignes sont ajoutées et les entrées sont triées selon les colonnes Team et Age, les deux en ordre ascendant.

6. Double-cliquez sur le premier tLogRow pour afficher sa vue Basic settings.
7. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour une meilleure lisibilité des résultats de tri.

**Agréger les données triées**

**Procédure**

1. Double-cliquez sur le **tAggregateSortedRow** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans la fenêtre qui s'ouvre, définissez le schéma en ajoutant cinq colonnes : **AggTeam** de type **String**, **AggCount**, **MinAge**, **MaxAge** et **AvgAge** de type **Integer**.
Cliquez sur OK pour fermer l’éditeur de schéma et acceptez la propagation proposée pour la boîte de dialogue.

3. Dans le champ Input rows count, saisissez le nombre exact de lignes de données d’entrée. Dans cet exemple, saisissez 8.

4. Cliquez sur le bouton [+] sous la table Group by pour ajouter autant de lignes que nécessaire et spécifier l’ensemble d’agrégation dans la table. Dans cet exemple, les données sont agrégées selon la colonne Team.

5. Cliquez sur le bouton [+] sous la table Operations pour ajouter autant de lignes que nécessaire. Spécifiez l’opération à effectuer, ainsi que la colonne d’entrée correspondant de laquelle prendre les données pour chaque colonne de sortie. Dans cet exemple, vous allez calculer le nombre d’entrées, l’âge minimal, l’âge maximal et l’âge moyen pour chaque équipe.

6. Double-cliquez sur le second tLogRow pour ouvrir sa vue Basic settings.

7. Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour une meilleure lisibilité des résultats du tri.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl + S afin de sauvegarder le Job.
2. Exécutez le Job en appuyant sur F6 ou en cliquant sur le bouton Run dans la vue Run.
Comme dans la capture d'écran ci-dessus, les entrées sont triées selon la colonne *Team*, puis selon la colonne *Age*, toutes deux en ordre ascendant. Les entrées triées sont ensuite agrégées selon la colonne *Team*. 
tAmazonAuroraClose

Ce composant ferme une connexion active à une instance d’une base de données Amazon Aurora afin de libérer des ressources.

Propriétés du tAmazonAuroraClose Standard

Ces propriétés sont utilisées pour configurer le tAmazonAuroraClose s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonAuroraClose Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Component List</td>
<td>Sélectionnez dans la liste le composant tAmazonAuroraConnection ouvrant la connexion à fermer.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case afin de collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option. Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
</tbody>
</table>

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec d’autres composants Amazon Aurora, notamment avec le tAmazonAuroraConnection et le tAmazonAuroraCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton &quot;+&quot; pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez Scénario : Gestion des données avec Amazon Aurora à la page 148.
tAmazonAuroraCommit

Ce composant commite en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes, et permet un gain de performance, en utilisant une connexion unique.

Le composant tAmazonAuroraCommit valide les données traitées par le Job dans la base de données connectée Amazon Aurora.

**Propriétés du tAmazonAuroraCommit Standard**

Ces propriétés sont utilisées pour configurer le tAmazonAuroraCommit s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonAuroraCommit Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component List</td>
<td>Sélectionnez le composant tAmazonAuroraConnection pour lequel vous souhaitez commiter l’action à effectuer.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Avertissement :**

Si vous utilisez un lien de type Row > Main pour relier le tAmazonAuroraCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.

**Advanced settings**

| tStatCatcher Statistics                       | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |
### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Amazon Aurora, notamment avec le *tAmazonAuroraConnection* et le *tAmazonAuroraRollback*. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste *Component List* de la vue *Basic settings* devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend. |
Scénario associé

Pour un scénario associé, consultez Scénario : Gestion des données avec Amazon Aurora à la page 148.
**tAmazonAuroraConnection**

Ce composant ouvre une connexion à une instance de la base de données pouvant être réutilisée par d'autres composants Amazon Aurora.

### Propriétés du tAmazonAuroraConnection Standard

Ces propriétés sont utilisées pour configurer le tAmazonAuroraConnection s'exécutant dans le framework de Jobs Standard.

Le composant tAmazonAuroraConnection Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Property Type</td>
<td>Peut être <strong>Built-In</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-In</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Saisissez l'adresse IP ou le nom de l'hôte de la base de données Amazon Aurora.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d'écoute de la base de données Amazon Aurora.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données à utiliser.</td>
</tr>
<tr>
<td>Additional JDBC parameters</td>
<td>Spécifiez des informations supplémentaires de connexion à la base de données créée.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d'authentification de l'utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
### Use or register a shared DB Connection

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfaînt ou parents.

Cette option est incompatible avec les options Use dynamic job et Use an independent process to run subjob du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.

Cette case est indisponible lorsque la case Specify a data source alias est cochée.

### Specify a data source alias

Cochez cette case et spécifiez l’alias de la source de données créée dans **Talend Runtime** pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans **Talend Runtime**. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans **Talend Runtime** pour récupérer les données d’une base de données MySQL à la page 2647.

Cette case n’est pas disponible lorsque la case Use or register a shared DB Connection est cochée.

### Data source alias

Saisissez l’alias de la source de données créée, du côté **Talend Runtime**.

Ce champ s’affiche lorsque la case Specify a data source alias est cochée.

### Advanced settings

#### Auto Commit

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commit pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour
gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

Variables globales

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable _After_ et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case _Die on error_ est décochée, si le composant a cette option.

Une variable _Flow_ fonctionne durant l’exécution d’un composant. Une variable _After_ fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches _Ctrl+ESpace_ pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le _Guide utilisateur du Studio Talend_.

Utilisation

Ce composant est généralement utilisé avec d’autres composants Amazon Aurora, notamment avec le _tAmazonAuroraCommit_ et le _tAmazonAuroraRollback_.

Scénario associé

Pour un scénario associé, consultez _Scénario : Gestion des données avec Amazon Aurora_ à la page 148.
tAmazonAuroraInput

Ce composant lit une base de données Amazon Aurora et en extrait des champs à l’aide de requêtes.
tAmazonAuroraInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux Row > Main.

Propriétés du tAmazonAuroraInput Standard

Ces propriétés sont utilisées pour configurer le tAmazonAuroraInput s’exécutant dans le framework de Jobs Standard.
Le composant tAmazonAuroraInput Standard appartient aux familles Cloud et Databases.
Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Peut être Built-In ou Repository.</td>
</tr>
<tr>
<td>Built-In</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Saisissez l’adresse IP ou le nom de l’hôte de la base de données Amazon Aurora.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute de la base de données Amazon Aurora.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données à utiliser.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td>Table Name</td>
<td>Saisissez le nom de la table à lire.</td>
</tr>
</tbody>
</table>
**Query Type et Query**

Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

**Guess Query**

Cliquez sur le bouton afin de générer la requête correspondant au schéma de la table dans le champ Query.

**Guess schema**

Cliquez sur ce bouton afin de récupérer le schéma de la table.

**Specify a data source alias**

Cochez cette case et spécifiez l’alias de la source de données créée dans Talend Runtime pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647.

Cette case est indisponible lorsque la case Use an existing connection est cochée.

**Data source alias**

Saisissez l’alias de la source de données créée du côté Talend Runtime.

Ce champ s’affiche uniquement lorsque la case Specify a data source alias est cochée.

### Advanced settings

**Additional JDBC parameters**

Spécifiez des informations supplémentaires de connexion à la base de données créée. Lorsque vous devez gérer des données de type 0000-00-00 00:00:00 à l’aide de ce composant, configurez le paramètre à noDatetimeStringSync=true&zeroDateTimeBehavior=convertToNull.

Ce champ est indisponible lorsque la case Use an existing connection est cochée dans l’onglet Basic settings.

**Enable stream**

Cochez cette case pour préférer le streaming à la mise en mémoire tampon, ce qui permet au code de lire les données d’une table volumineuse sans consommer beaucoup de mémoire, afin d’optimiser les performances.

**Trim all the String/Char columns**

Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.

**Trim column**

Cochez la (les) case(s) de la colonne Trim afin de supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.

L’option est indisponible lorsque la case Trim all the String/Char columns est cochée.
tAmazonAuroraInput

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant. |

Global Variables

| Global Variables | NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. |
| QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères. |
| ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
| Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |
| Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
| Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début dans un Job ou sous-job et nécessite un lien de sortie. |
| Dynamic settings | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. |
| La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. |
| Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à |
**Scénario : Gestion des données avec Amazon Aurora**

Ce scénario décrit un Job écrivant des informations relatives à des utilisateurs dans Amazon Aurora, lisant ces information dans Amazon Aurora et les affichant dans la console.

Le scénario requiert les sept composants suivants :
- un **tAmazonAuroraConnection** : ouvre une connexion à Amazon Aurora.
- un **tFixedFlowInput** : définit la structure des données et envoie ces données au composant suivant.
• un **tAmazonAuroraOutput** : écrit les données reçues du composant précédant dans Amazon Aurora.
• un **tAmazonAuroraCommit** : commite en une fois les données traitées dans Amazon Aurora.
• un **tAmazonAuroraInput** : lit les données dans Amazon Aurora.
• un **tLogRow** : affiche les données reçues du composant précédant dans la console.
• un **tAmazonAuroraClose** : ferme la connexion à Amazon Aurora.

**Construire le Job**

**Procédure**

1. Créez un nouveau Job et ajoutez les sept composants précédemment listés en saisissant leur nom dans l’espace de modélisation graphique ou en les glissant depuis la **Palette**.
2. Connectez le **tFixedFlowInput** au **tAmazonAuroraOutput** à l’aide d’un lien **Row > Main**.
3. Répétez l’opération afin de connecter le **tAmazonAuroraInput** au **tLogRow**.
4. Connectez le **tAmazonAuroraConnection** au **tFixedFlowInput** à l’aide d’un lien **Trigger > OnSubjobOk**.
5. Répétez l’opération pour connecter le **tFixedFlowInput** au **tAmazonAuroraCommit**, le **tAmazonAuroraCommit** au **tAmazonAuroraInput** et le **tAmazonAuroraInput** au **tAmazonAuroraClose**.

**Configurer les composants**

**Ouvrir une connexion à Amazon Aurora**

**Procédure**

1. Double-cliquez sur le **tAmazonAuroraConnection** pour ouvrir sa vue **Basic settings**.

![Image du tAmazonAuroraConnection](image)

2. Dans les champs **Host**, **Port**, **Database**, **Username** et **Password**, saisissez les informations requises pour la connexion à Amazon Aurora.

**Écrire les données dans Amazon Aurora**

**Procédure**

1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.
2. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la fenêtre qui s’ouvre, définissez le schéma en ajoutant trois colonnes : id de type Integer, name et city de type String.

Cliquez sur OK afin de valider les modifications et acceptez la propagation proposées par la boîte de dialogue.

3. Dans la zone Mode, sélectionnez Use Inline Content (delimited file) et saisissez les informations relatives aux utilisateurs dans le champ Content.

```
1;George;Bismarck
2;Abraham;Boise
3;Taylor;Nashville
4;William;Jefferson City
5;Kate;Jackson
6;James;Boise
7;Gerald;Little Rock
8;Chester;Richmond
9;Thomas;Springfield
10;John;Nashville
```

4. Double-cliquez sur le tAmazonAuroraOutput pour ouvrir sa vue Basic settings.
5. Cochez la case **Use an existing connection** et, dans la liste **Component List** qui s’affiche, sélectionnez le composant de connexion que vous avez configuré.

6. Dans le champ **Table**, saisissez le nom de la table ou sélectionnez la table dans laquelle vous souhaitez écrire les données. Dans cet exemple, saisissez **TalendUser**.

7. Sélectionnez **Drop table if exists and create** dans la liste **Action on table** et sélectionnez **Insert** dans la liste **Action on data**.

8. Double-cliquez sur le **tAmazonAuroraCommit** pour ouvrir sa vue **Basic settings**.

9. Décochez la case **Close Connection** si elle est sélectionnée.

**Récupérer les données d’Amazon Aurora**

**Procédure**

1. Double-cliquez sur le **tAmazonAuroraInput** pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Use an existing connection** et, dans la liste **Component List** qui s’affiche, sélectionnez le composant de connexion que vous avez configuré.
3. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la fenêtre qui s'ouvre, définissez le schéma en ajoutant trois colonnes : id de type Integer, name et city de type String. La structure des données est la même que la structure définie pour le tFixedFlowInput.

4. Dans le champ Table Name, saisissez le nom ou sélectionnez la tables dans laquelle écrire les données. Dans cet exemple, saisissez TalendUser.

5. Cliquez sur le bouton Guess Query afin de générer la requête. Le champ Query est renseigné avec la requête automatiquement générée.

6. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings.

7. Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour une lisibilité optimale des résultats.

Fermer la connexion à Amazon Aurora

Procédure

1. Double-cliquez sur le tAmazonAuroraClose pour ouvrir sa vue Basic settings.

2. Dans la liste Component List, sélectionnez le composant de connexion que vous avez configuré.

Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur les touches Ctrl + S afin de sauvegarder le Job.
2. Appuyez sur F6 ou cliquez sur le bouton Run de l’onglet Run pour exécuter le Job.
Comme affiché ci-dessus, les informations relatives aux utilisateurs sont écrites dans Amazon Aurora, puis les données sont récupérées depuis Amazon Aurora et affichées dans la console.

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>city</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>George</td>
<td>Bismarck</td>
</tr>
<tr>
<td>2</td>
<td>Abraham</td>
<td>Boise</td>
</tr>
<tr>
<td>3</td>
<td>Taylor</td>
<td>Nashville</td>
</tr>
<tr>
<td>4</td>
<td>William</td>
<td>Jefferson City</td>
</tr>
<tr>
<td>5</td>
<td>Alexander</td>
<td>Jackson</td>
</tr>
<tr>
<td>6</td>
<td>James</td>
<td>Boise</td>
</tr>
<tr>
<td>7</td>
<td>Gerald</td>
<td>Little Rock</td>
</tr>
<tr>
<td>8</td>
<td>Tony</td>
<td>Richmond</td>
</tr>
<tr>
<td>9</td>
<td>Thomas</td>
<td>Springfield</td>
</tr>
<tr>
<td>10</td>
<td>Andre</td>
<td>Nashville</td>
</tr>
</tbody>
</table>
tAmazonAuroraOutput

Ce composant écrit, met à jour, modifie ou supprime des entrées dans une base de données Amazon Aurora.

tAmazonAuroraOutput exécute l’action définie sur la table et/ou sur les données contenues dans la table, en fonction du flux entrant du composant précédent dans le Job.

Propriétés du tAmazonAuroraOutput Standard

Ces propriétés sont utilisées pour configurer le tAmazonAuroraOutput s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonAuroraOutput Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Peut être Built-In ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Saisissez l’adresse IP ou le nom de l’hôte de la base de données Amazon Aurora.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Saisissez le numéro du port d’écoute de la base de données Amazon Aurora.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez le nom de la base de données à utiliser.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez les informations d’authentification de l’utilisateur à la base de données.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Saisissez le nom de la table à écrire. Notez qu’une table peut être écrite à la fois.</td>
</tr>
<tr>
<td><strong>Action on table</strong></td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td>• <strong>None</strong> : n’effectuer aucune opération sur la table.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Drop and create table</strong> : supprimer la table puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Create table</strong> : créer une table qui n’existe pas encore.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Create table if not exists</strong> : créer la table si nécessaire.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Drop table if exists and create</strong> : supprimer la table si elle existe déjà, puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Clear table</strong> : supprimer le contenu de la table.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Truncate table</strong> : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.</td>
</tr>
<tr>
<td><strong>Action on data</strong></td>
<td>Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td>• <strong>Insert</strong> : ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update</strong> : mettre à jour les entrées existantes.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Insert or update</strong> : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update or insert</strong> : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.</td>
</tr>
</tbody>
</table>
- **Delete**: supprimer les entrées correspondant au flux d’entrée.

- **Replace**: ajouter de nouvelles entrées à la table. Si une ancienne ligne dans la table a la même valeur qu’une nouvelle ligne pour un index PRIMARY KEY ou UNIQUE, l’ancienne ligne est supprimée avant insertion de la nouvelle.

- **Insert or update on duplicate key or unique index**: ajouter des entrées si la valeur insérée n’existe pas ou mettre à jour les entrées si la valeur insérée existe déjà et qu’un risque de violation d’une clé unique se présente.

- **Insert ignore**: ajouter uniquement de nouvelles lignes afin d’empêcher les erreurs de doublons de clés.

⚠️ **Avertissement**:

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update et Delete**. Pour cela, cliquez sur le bouton `[...]` à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

### Built-In

Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

### Repository

Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Specify a data source alias**

Cochez cette case et spécifiez l'alias de la source de données crée dans **Talend Runtime** pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans **Talend Runtime**. Pour voir un exemple de cas d’utilisation, consultez **Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647**.

Cette case est indisponible lorsque la case **Use an existing connection** est cochée.

**Data source alias**

Saisissez l’alias de la source de données crée du côté **Talend Runtime**.

Ce champ s’affiche uniquement lorsque la case **Specify a data source alias** est cochée.

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

**Advanced settings**

**Additional JDBC parameters**

Spécifiez des informations supplémentaires de connexion à la base de données crée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.

**Remarque :**

Vous pouvez appuyer sur **Ctrl+Espace** afin d’accéder à une liste de variables globales prédéfinies.
| **Extend Insert** | Cochez cette case pour insérer un ensemble de lignes définies au lieu d’insérer les lignes une par une. Cette option permet un important gain de performance. Cette case est disponible lorsque l’option *Insert* est sélectionnée dans la liste *Action on data*, dans l’onglet *Basic settings*.  

**Remarque :**
Cette option n’est pas compatible avec le lien *Reject*. Vous devez donc la décocher si vous utilisez un lien *Reject* en sortie du composant. |
| **Number of rows per insert** | Saisissez le nombre de lignes à insérer par opération. Notez que, plus haute est la valeur spécifiée, plus la performance est basse, à cause de la consommation de mémoire. Ce champ est disponible lorsque la case *Extend Insert* est cochée. |
| **Use Batch** | Cochez cette case pour activer le mode de traitement par lots pour le traitement des données. Cette case est disponible lorsque vous sélectionnez *Update* ou *Delete* dans la liste *Action on data*, dans la vue *Basic settings*. |
| **Batch Size** | Spécifiez le nombre d’enregistrements à traiter dans chaque lot. Ce champ est disponible uniquement lorsque la case *Use batch mode* est cochée. |
| **Commit every** | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution. |
| **Additional columns** | Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier. Cliquez sur le bouton [+ ] sous la table pour ajouter des colonnes et configurez les paramètres suivants pour chaque colonne.  

- **Name** : saisissez le nom de la colonne du schéma à modifier ou insérer.  
- **SQL expression** : saisissez l’instruction SQL à exécuter afin de modifier ou insérer les données dans la colonne correspondante.  
- **Position** : sélectionnez Before, After ou Replace, selon l’action à effectuer sur la colonne de référence.  
- **Reference column** : saisissez une colonne de référence pouvant être utilisée par le *tAmazonAuroraOutput* pour localiser ou remplacer la nouvelle colonne ou la colonne à modifier. |
| Use field options | Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.  
- **Key in update** : cochez la case de la colonne correspondante, par rapport à laquelle les données sont mises à jour.  
- **Key in delete** : cochez la case de la colonne correspondante, par rapport à laquelle les données sont supprimées.  
- **Updatable** : cochez la case si les données de la colonne correspondante peuvent être mises à jour.  
- **Insertable** : cochez la case si les données de la colonne correspondante peuvent être insérées. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Hint Options</td>
<td>Cochez cette case pour configurer les indicateurs pouvant permettre d’optimiser l’exécution d’une requête.</td>
</tr>
</tbody>
</table>
| Hint Options | Cochez cette case pour activer la zone de configuration des indicateurs (ou Hints) permettant d’optimiser l’exécution d’une requête. Dans cette zone, les paramètres sont :  
- **HINT** : spécifiez l’indicateur dont vous avez besoin, en utilisant la syntaxe /*+ */.  
- **POSITION** : spécifiez la place de l’indicateur dans une instruction SQL.  
- **SQL STMT**: sélectionnez l’instruction SQL INSERT, UPDATE, ou DELETE à utiliser. |
| Enable debug mode | Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données. |
| Use duplicate key update mode insert | Cochez cette case pour activer le mode ON DUPLICATE KEY UPDATE, puis cliquez sur le bouton [*] sous la table pour ajouter des colonnes à mettre à jour et spécifiez l’action de mise à jour à effectuer sur la colonne correspondante.  
- **Column** : saisissez le nom de la colonne à mettre à jour.  
- **Value** : saisissez l’action à effectuer sur la colonne.  
Cette case est disponible lorsque l’option **Insert** est sélectionnée dans la liste **Action on data** de la vue **Basic settings**. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant. |

### Global Variables

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. |
### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de sortie. Il vous permet d’effectuer des actions sur les données d’une table dans une base de données Amazon Aurora. Il permet également de créer un flux de rejet, à l’aide d’un lien Row > Rejects afin de filtrer les données en erreur. Pour un scénario similaire, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675. |
| Dynamic settings | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. |
Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Scénario associé

Pour un scénario associé, consultez Scénario : Gestion des données avec Amazon Aurora à la page 148.
tAmazonAuroraRollback

Ce composant vous permet d'annuler des modifications apportées à la base de données Amazon Aurora pour éviter le commit de transaction involontaire si une erreur survient.

Propriétés du tAmazonAuroraRollback Standard

Ces propriétés sont utilisées pour configurer le tAmazonAuroraRollback s'exécutant dans le framework de Jobs Standard.

Le composant tAmazonAuroraRollback Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

| Database | Sélectionnez un type de base de données dans la liste et cliquez sur Apply. |
| Component List | Sélectionnez le composant tAmazonAuroraConnection pour lequel effectuer l’action de rollback. |
| Close Connection | Cette case est cochée par défaut est vous permet de fermer la connexion à la base de données une fois le rollback effectué. Décochez cette case pour continuer à utiliser la connexion sélectionnée une fois sa tâche terminée. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

Global Variables

| Global Variables | ERROR MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

**Règle d'utilisation**

Ce composant est généralement utilisé avec d'autres composants Amazon Aurora, notamment avec le tAmazonAuroraConnection et le tAmazonAuroraCommit.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tAmazonEMRListInstances

Ce composant liste les détails concernant les groupes d’instances dans un cluster, dans Amazon EMR (Elastic MapReduce).

# Propriétés du tAmazonEMRListInstances Standard

Ces propriétés sont utilisées pour configurer le tAmazonEMRListInstances s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonEMRListInstances Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

## Basic settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access key et Secret key</strong></td>
<td>Spécifiez les clés d’accès (l’ID de la clé d’accès dans le champ <strong>Access Key</strong> et la clé secrète d’accès dans le champ <strong>Secret Key</strong>) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés d’accès d’AWS, consultez Clés d’accès (ID de clé d’accès et clé d’accès secrète). Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ <strong>Secret key</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Inherit credentials from AWS role</strong></td>
<td>Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
<tr>
<td><strong>Assume role</strong></td>
<td>Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.</td>
</tr>
<tr>
<td>• <strong>Role ARN</strong></td>
<td>nom Amazon Resource Name (ARN) du rôle.</td>
</tr>
<tr>
<td>• <strong>Role session name</strong></td>
<td>identifiant de la session du rôle.</td>
</tr>
<tr>
<td>• <strong>Session duration (minutes)</strong></td>
<td>durée (en minutes) pour laquelle est active la session du rôle.</td>
</tr>
<tr>
<td><strong>Region</strong></td>
<td>Spécifiez la région AWS en sélectionnant dans la liste un nom de région ou saisissez un nom de région entre guillemets doubles (par exemple “us-east-1”). Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).</td>
</tr>
</tbody>
</table>
d’informations concernant comment spécifier la région AWS, consultez Choix d’une région AWS (en anglais).

**Filter master and core instances**
Cochez cette case pour ignorer les groupes d’instances Core et Master et lister uniquement les groupes d’instances de tâches.

**Cluster id**
Saisissez l’ID du cluster pour lequel vous souhaitez lister les groupes d’instances.

### Advanced settings

**STS Endpoint**
Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.
Cette case est disponible uniquement lorsque la case **Assume role** est cochée.

**tStatCatcher Statistics**
Cochez cette case afin de collecter les informations de log au niveau du Job, ainsi qu’au niveau de chaque composant.

### Global Variables

**Global Variables**

- **CURRENT_GROUP_ID**: ID du groupe d’instances courant. Cette variable est une variable After et retourne une chaîne de caractères.
- **CURRENT_GROUP_NAME**: nom du groupe d’instances courant. Cette variable est une variable After et retourne une chaîne de caractères.
- **ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**

Le tAmazonEMRListInstances est généralement utilisé en tant que composant de début dans un Job ou un sous-job.
Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tAmazonEMRManage

Ce composant lance ou termine un cluster sur Amazon EMR (Elastic MapReduce).

Propriétés du tAmazonEMRManage Standard

Ces propriétés sont utilisées pour configurer le tAmazonEMRManage s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonEMRManage Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

| Access key et Secret key | Spécifiez les clés d’accès (l’ID de la clé d’accès dans le champ Access Key et la clé secrète d’accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés d’accès d’AWS, consultez Clés d’accès (ID de clé d’accès et clé d’accès secrète).

Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres. |
| Inherit credentials from AWS role | Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2. |
| Assume role | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.

- **Role ARN** : nom Amazon Resource Name (ARN) du rôle.
- **Role session name** : identifiant de la session du rôle.
- **Session duration (minutes)** : durée (en minutes) pour laquelle est active la session du rôle.

Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais). |
| Action | Sélectionnez dans la liste une action à effectuer, Start ou Stop.

- **Start** : lance un cluster EMR. |
### Stop
- **Termine un cluster Amazon EMR.**

### Region
- Spécifiez la région AWS en sélectionnant dans la liste un nom de région ou saisissez un nom de région entre guillemets doubles (par exemple "us-east-1"). Pour plus d’informations concernant comment spécifier la région AWS, consultez [Choix d’une région AWS](#).

### Cluster name
- **Saisissez le nom du cluster.**

### Cluster version
- Sélectionnez la version du cluster.
- Vous pouvez également cocher la case **Customize Version and Application** dans la vue **Advanced settings** afin de personnaliser les informations de version.
- Cette propriété n’est pas disponible lorsque la case **Customize Version and Application** est cochée.

### Application
- Sélectionnez les applications à installer sur le cluster.
- Vous pouvez également cocher la case **Customize Version and Application** dans la vue **Advanced settings** afin de personnaliser les informations des applications.
- Cette propriété est disponible lorsqu’une version d’EMR est sélectionnée dans la liste **Cluster version** et que la case **Customize Version and Application** est décochée.

### Service role
- **Saisissez le rôle IAM (Identity and Access Management) pour le service Amazon EMR. Le rôle par défaut est EMR_DefaultRole. Pour utiliser ce rôle par défaut, vous devez l’avoir déjà créé.**

### Job flow role
- **Saisissez le rôle IAM pour les instances EC2 gérées par Amazon EMR. Le rôle par défaut est EMR_EC2_DefaultRole. Pour utiliser ce rôle par défaut, vous devez l’avoir déjà créé.**

### Enable log
- Cochez cette case pour activer l’enregistrement des logs et, dans le champ qui s’affiche, spécifiez le chemin d’accès à un dossier dans un bucket S3 dans lequel vous souhaitez qu’Amazon EMR écrive les données de log.

### Use EC2 key pair
- Cochez cette case pour associer une paire de clés Amazon EC2 (Elastic Compute Cloud) au cluster et, dans le champ affiché, saisissez le nom de votre paire de clés EC2.

### Predicate
- Spécifiez le(s) cluster(s) que vous souhaitez arrêter :
  - **All running clusters** : tous les clusters en cours d’exécution seront arrêtés.
  - **All running clusters with predefined name** : le cluster en cours d’exécution ayant le nom défini sera arrêté. Dans le champ **Cluster name** affiché, vous devez spécifier le nom du cluster à arrêter.
  - **Running cluster with predefined id** : le cluster en cours d’exécution ayant un ID donné va être...
arrêté. Dans le champ **Cluster id** affiché, vous devez spécifier l’ID du cluster à arrêter. Cette liste est disponible uniquement lorsque l’option **Stop** est sélectionnée dans la liste **Action**.

| Instance count | Saisissez le nombre d’instances d’Amazon EC2 à initialiser. |
| Master instance type | Sélectionnez le type d’instance maître à initialiser. |
| Slave instance type | Sélectionnez le type d’instance esclave à initialiser. |

**Advanced settings**

| STS Endpoint | Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées. Cette case est disponible uniquement lorsque la case **Assume role** est cochée. |
| Wait for cluster ready | Cochez cette case pour laisser votre Job attendre jusqu’à ce que le lancement du cluster soit terminé. |
| Visible to all users | Cochez cette case pour rendre visible le cluster à tous les utilisateurs IAM. |
| Termination Protect | Cochez cette case pour activer la protection contre les arrêts afin d’empêcher les instances de s’éteindre à cause d’erreurs ou de problèmes au cours du traitement. |
| Enable debug | Cochez cette case pour activer le mode débogage. |

**Customize Version and Application**

Cochez cette case pour personnaliser la version du cluster et les applications à installer sur le cluster.

- **Cluster version** : saisissez la version du cluster.
- **Applications** : cliquez sur le bouton [+] sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour une application. Spécifiez l’application en cliquant du côté droit de la cellule et en sélectionnant l’application dans la liste déroulante qui s’affiche, ou en saisissant le nom de l’application dans la cellule si elle n’est pas dans la liste.

**Subnet id**

Spécifiez l’identifiant du sous-réseau de Amazon VPC (Virtual Private Cloud) dans lequel vous souhaitez démarrer le flux du Job.

**Availability Zone**

Spécifiez la zone de disponibilité pour les instances EC2 de votre cluster.

**Master security group**

Spécifiez le groupe de sécurité pour l’instance maître.
| Additional master security groups | Spécifiez des groupes de sécurité supplémentaires pour l’instance maître et séparez-les à l’aide d’une virgule, par exemple, `gname1, gname2, gname3`. |
| Slave security group | Spécifiez le groupe de sécurité pour les instances esclaves. |
| Additional slave security groups | Spécifiez des groupes de sécurité supplémentaires pour les instances esclaves et séparez-les à l’aide d’une virgule, par exemple, `gname1, gname2, gname3`. |
| Service Access Security Group | Spécifiez l’identifiant du groupe de sécurité Amazon EC2 pour que le service Amazon EMR accède aux clusters du Cloud privé virtual via un sous-réseau privé. Pour plus d’informations concernant la création d’un sous-réseau privé pour activer le groupe de sécurité d’accès au service dans Amazon EMR, consultez Scénario 2 : VPC avec des sous-réseaux publics et privés (NAT). |
| Actions | Spécifiez les actions de bootstrapping associées au cluster, en cliquant sur le bouton `[+]` sous la table, pour ajouter autant de lignes que nécessaire, chaque ligne pour une action de bootstrapping. Configurez les paramètres pour chaque action :
  - **Name** : saisissez le nom de l’action de bootstrapping.
  - **Script location** : spécifiez l’emplacement du script exécuté par l’action de bootstrapping, par exemple, `s3://ap-northeast-1.elasticmapreduce/bootstrap-actions/run-if`.
  - **Arguments** : saisissez la liste des arguments de commandes (séparés par des virgules) passés à l’action de bootstrapping, par exemple, `"arg0","arg1","arg2"`. Pour plus d’informations concernant les actions de bootstrapping, consultez BootstrapActionConfig (en anglais). |
| Steps | Spécifiez les étapes du flux du job à invoquer sur le cluster après son lancement, en cliquant sur le bouton `[+]` pour ajouter autant de lignes que nécessaire, chaque ligne pour une étape. Configurez les paramètres suivants pour chaque étape :
  - **Name** : saisissez le nom de l’étape du flux du job.
  - **Action on Failure** : cliquez dans la cellule et, dans la liste déroulante, sélectionnez l’action à effectuer si l’étape du flux du job échoue.
  - **Main Class** : saisissez le nom de la classe principale dans le fichier Java spécifié. Si aucun nom n’est spécifié, le fichier Jar doit spécifier une classe principale dans son fichier manifest.
  - **Jar** : saisissez le chemin d’accès au fichier Jar exécuté au cours de l’étape, par exemple, `s3://inputjar/test.jar` |
• **Args** : saisissez la liste des arguments de l’invite de commande (séparés par une virgule) passés à la fonction principale du fichier Jar lors de l’exécution, par exemple ’arg0’,”arg1”,”arg2”.

Pour plus d’informations concernant les étapes du flux du job, consultez StepConfig (en anglais).

**Keep alive after steps complete**

Cochez cette case pour garder actif le flux du job une fois les étapes terminées.

**Wait for steps to complete**

Cochez cette case pour que votre Job attende jusqu’à ce que les étapes du flux du job soient terminées.

Cette case est disponible uniquement lorsque la case *Wait for cluster ready* est cochée.

**Properties**

Spécifiez les informations de classification et de propriété fournies à l’objet de configuration du cluster EMR à créer, en cliquant sur le bouton [+] sous la table pour ajouter autant que lignes que nécessaire, chaque ligne pour une propriété. Configurez les paramètres suivants :

- **Classification** : spécifiez la classification de la configuration.
- **Key** : saisissez la clé de la propriété.
- **Value** : saisissez la valeur de la propriété.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLUSTER_FINAL_ID</td>
<td>ID du cluster. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>CLUSTER_FINAL_NAME</td>
<td>Nom du cluster. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

| Règle d’utilisation | Le tAmazonEMRManage est généralement utilisé en tant que composant standalone. |

### Gérer un cluster Amazon EMR

Voici un exemple d’utilisation des composants Talend pour gérer un cluster Amazon EMR.
**Créer un Job de gestion de cluster Amazon EMR**

Créez un Job démarrant un nouveau cluster Amazon EMR, redimensionnez le cluster, puis listez les informations des ID et noms des groupes d’instance dans le cluster.

**Procédure**

1. Créez un nouveau Job et ajoutez un `tAmazonEMRManage`, un `tAmazonEMRResize`, un `tAmazonEMRListInstances` et un `tJava`, en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la `Palette`.
2. Reliez le `tAmazonEMRManage` au `tAmazonEMRResize`, à l’aide d’un lien `Trigger > OnSubjobOk`.
3. Reliez le composant `tAmazonEMRResize` au `tAmazonEMRListInstances` à l’aide d’un lien `Trigger > OnSubjobOk`.
4. Reliez le composant `tAmazonEMRListInstances` au `tJava`, à l’aide d’un lien `Row > Iterate`.

**Démarrer un nouveau cluster Amazon EMR**

Configurez le `tAmazonEMRManage` pour qu’il démarre un nouveau cluster Amazon EMR.

**Procédure**

1. Double-cliquez sur le `tAmazonEMRManage` pour ouvrir sa vue `Basic settings`. 
2. Dans les champs Access Key et Secret Key, saisissez les informations d’authentification requises pour accéder à Amazon S3.

3. Dans la liste Action, sélectionnez Start pour démarrer un cluster.

4. Sélectionnez la région AWS dans la liste Region. Dans cet exemple, sélectionnez Asia Pacific (Tokyo).


7. Cochez la case Enable log et, dans le champ qui s’affiche, spécifiez le chemin d’accès au dossier, dans un bucket S3, où vous souhaitez qu’Amazon EMR écrive les données de log. Dans cet exemple, saisissez s3://talend-doc-emr-bucket.

**Redimensionner le cluster Amazon EMR en ajoutant un nouveau groupe d’instance de tâche**

Configurez le composant tAmazonEMRResize pour redimensionner un cluster Amazon EMR en cours de fonctionnement en ajoutant un nouveau groupe d’instance de tâche.

**Procédure**

1. Double-cliquez sur le tAmazonEMRResize pour ouvrir sa vue Basic settings.
2. Dans les champs **Access Key** et **Secret Key**, saisissez les informations d’authentification requises pour accéder à Amazon S3.

3. Dans la liste déroulante **Action**, sélectionnez **Add task instance group** afin de redimensionner le cluster en ajoutant un nouveau groupe d’instance de tâche.

4. Dans le champ **Cluster id**, saisissez l’ID du cluster à redimensionner. Dans cet exemple, la valeur retournée de la variable globale `CLUSTER_FINAL_ID` du précédent tAmazonEMRManage est utilisée. Notez que vous pouvez récupérer la variable globale en appuyant sur les touches **Ctrl + Espace** et en sélectionnant la variable globale adéquate dans la liste.


6. Dans le champ **Instance count**, spécifiez le nombre d’instances à créer.

7. Dans la liste **Task instance type**, sélectionnez le type d’instances à créer.

### Lister les groupes d’instances dans le cluster Amazon EMR

Configurez le tAmazonEMRListInstances et le tJava pour récupérer et afficher les informations d’ID et de nom de tous les groupes d’instances d’un cluster en cours d’exécution.

**Procédure**

1. Double-cliquez sur le tAmazonEMRListInstances pour ouvrir sa vue **Basic settings**.

![tAmazonEMRListInstances](image)

2. Dans les champs **Access Key** et **Secret Key**, saisissez les informations d’authentification requises pour accéder à Amazon S3.

3. Sélectionnez la région AWS dans la liste **Region**. Dans cet exemple, saisissez *Asia Pacific (Tokyo)*.

4. Décchez la case **Filter master and core instances** pour lister tous les groupes d’instances, y compris les groupes d’instances de type Master, Core et Task.

5. Dans le champ **Cluster id**, saisissez l’ID du cluster pour lequel lister les groupes d’instances. Dans cet exemple, la valeur retournée de la variable globale `CLUSTER_FINAL_ID` du précédent tAmazonEMRManage est utilisée.

6. Double-cliquez sur le composant tJava pour ouvrir sa vue **Basic settings**.

![tJava](image)

```java
System.out.println("----- Instance Groups -----");
System.out.println("Instance Group ID: "+
(String)globalMap.get("tAmazonEMRListInstances_1_CURRENT_GROUP_ID");
System.out.println("Instance Group Name: "+
(String)globalMap.get("tAmazonEMRListInstances_1_CURRENT_GROUP_NAME");
```
7. Dans le champ **Code**, saisissez le code suivant pour afficher les informations d'ID et de nom de chaque groupe d'instance dans le cluster.

```java
System.out.println("
===== Instance Group =====
Instance Group ID:    " + (String)globalMap.get("tAmazonEMRListInstances_1_CURRENT_GROUP_ID");
System.out.println("Instance Group Name:  " + (String)globalMap.get("tAmazonEMRListInstances_1_CURRENT_GROUP_NAME"));
```

**Exécuter le Job pour gérer le cluster Amazon EMR**

Après avoir configuré le Job et les composants utilisés dans le Job pour gérer le cluster Amazon EMR, vous pouvez exécuter le Job et vérifier les résultats d'exécution du Job.

**Procédure**

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job et appuyez sur **F6** pour l'exécuter.

   ```
   Starting job DOC17011_tAmazonEMRManage at 16:50 24-08-2016.
   [statistics] connecting to socket on port 3612
   [statistics] connected
   Waiting for cluster to become available.
   .................................................................
   .................................................................
   .................................................................
   ----- Instance Group -----
   Instance Group ID:  ig-3UHY1W2EZV9I
   Instance Group Name: master
   ----- Instance Group -----
   Instance Group ID:  ig-17BB2U9CI4X2
   Instance Group Name: talend-doc-instance-group
   ----- Instance Group -----
   Instance Group ID:  ig-3HLYDZP9CD33
   Instance Group Name: slave
   [statistics] disconnected
   Job DOC17011_tAmazonEMRManage ended at 17:00 24-08-2016. [exit code=0]
   ```

   Comme affiché ci-dessus, le Job démarre et redimensionne le cluster Amazon EMR, puis liste tous les groupes d'instances dans le cluster.

2. Consultez les détails du cluster dans la page Amazon EMR Cluster List afin de valider les résultats d'exécution du Job.
tAmazonEMRResize

Ce composant ajoute ou redimensionne un groupe d'instances de tâches dans un cluster, dans Amazon EMR (Elastic MapReduce).

Propriétés du tAmazonEMRResize Standard

Ces propriétés sont utilisées pour configurer le tAmazonEMRResize s'exécutant dans le framework de Jobs Standard.

Le composant tAmazonEMRResize Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

| Access key et Secret key | Spécifiez les clés d'accès (l'ID de la clé d'accès dans le champ Access Key et la clé secrète d'accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d'informations concernant les clés d'accès d'AWS, consultez Clés d'accès (ID de clé d'accès et clé d'accès secrète).

Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s'exécuter dans Amazon EC2 ou d'autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
</tbody>
</table>
| Assume role | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.

• Role ARN : nom Amazon Resource Name (ARN) du rôle.

• Role session name : identifiant de la session du rôle.

• Session duration (minutes) : durée (en minutes) pour laquelle est active la session du rôle.

Pour plus d'informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais). |
| Action | Sélectionnez dans la liste une action à effectuer.

• Add task instance group : ajouter un groupe d'instances de tâches dans un cluster. |
- **Resize task instance group** : redimensionner un groupe d'instances de tâches dans un cluster.

| **Region** | Spécifiez la région AWS en sélectionnant dans la liste un nom de région ou saisissez un nom de région entre guillemets doubles (par exemple "us-east-1"). Pour plus d'informations concernant comment spécifier la région AWS, consultez [Choix d'une région AWS](#). |
| **Cluster id** | Saisissez l'ID du cluster à redimensionner. |
| **Group name** | Saisissez le nom du groupe d'instances de tâches à ajouter. Ce champ est disponible uniquement lorsque l'action **Add task instance group** est sélectionnée dans la liste **Action**. |
| **Group id** | Saisissez l'ID du groupe d'instances de tâches à redimensionner. Ce champ est disponible uniquement lorsque l'action **Resize task instance group** est sélectionnée dans la liste **Action**. |
| **Instance count** | Saisissez le nombre d'instances pour le groupe d'instances de tâches. |
| **Task instance type** | Sélectionnez dans la liste déroulante un type d'instance pour toutes les instances dans le groupe d'instances de tâches à ajouter. Cette liste est disponible uniquement lorsque l'action **Add task instance group** est sélectionnée dans la liste **Action**. |
| **Request spot** | Cochez cette case pour lancer des instances Spot et, dans le champ **Bid price($)** affiché, saisissez le taux horaire maximal (en dollars), que vous souhaitez payer par instance. Cette case est disponible uniquement lorsque l'option **Add task instance group** est sélectionnée dans la liste **Action**. |

**Advanced settings**

| **STS Endpoint** | Cochez cette case et, dans le champ qui s'affiche, spécifiez l'endpoint du service AWS Security Token Service duquel les informations d'authentification sont récupérées. Cette case est disponible uniquement lorsque la case **Assume role** est cochée. |
| **tStatCatcher Statistics** | Cochez cette case afin de collecter les informations de log au niveau du Job, ainsi qu'au niveau de chaque composant. |
Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>TASK_GROUP_ID : ID du groupe d’instances de la tâche. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TASK_GROUP_NAME : nom du groupe d’instances de la tâche. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Utilisation

| Règle d’utilisation | Le tAmazonEMRResize est généralement utilisé en tant que composant standalone. |

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tAmazonMysqlClose

Ce composant ferme la connexion à une base de données connectée.

Propriétés du tAmazonMysqlClose Standard

Ces propriétés sont utilisées pour configurer le tAmazonMysqlClose s'exécutant dans le framework de Jobs Standard.

Le composant tAmazonMysqlClose Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tAmazonMysqlConnection dans la liste.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables

**Global Variables**

| ERROR_MESSAGE | message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant doit être utilisé avec des composants AmazonMysql, notamment avec le tAmazonMysqlCommit et le tAmazonMysqlConnection.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tAmazonMysqlCommit

Ce composant commite en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes, et permet un gain de performance, en utilisant une connexion unique.

Le composant tAmazonMysqlCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tAmazonMysqlCommit Standard

Ces propriétés sont utilisées pour configurer le tAmazonMysqlCommit s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonMysqlCommit Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tAmazonMysqlConnection dans la liste.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décrochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. Avertissement : Si vous utilisez un lien de type Row &gt; Main pour relier le tAmazonMysqlCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
### Global Variables


### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants AmazonMysql et notamment avec les composants tAmazonMysqlConnection et tAmazonMysqlRollback.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez  Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
**Scénario associé**

Pour un scénario associé au composant **tAmazonMysqlCommit**, consultez **Scénario : Insérer des données dans des tables mère/fille** à la page 2620.
tAmazonMysqlConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Le tAmazonMysqlConnection ouvre une connexion vers une base de données afin d’effectuer une transaction.

**Propriétés du tAmazonMysqlConnection Standard**

Ces propriétés sont utilisées pour configurer le tAmazonMysqlConnection s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonMysqlConnection Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

*i* Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez [Composants de bases de données dynamiques](#) à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>DB Version</td>
<td>MySQL 5 est disponible.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Additional JDBC parameters</td>
<td>Spécifiez des informations supplémentaires de connexion à la base de données créée.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue</td>
</tr>
</tbody>
</table>
qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Use or register a shared DB Connection**

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ *Shared DB Connection Name* qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options *Use dynamic job* et *Use an independent process to run subjob* du composant *tRunJob*. Utiliser une connexion partagée avec un composant *tRunJob* ayant une de ces options activée fera échouer votre Job.

---

**Advanced settings**

**Auto Commit**

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne committe pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

---

**Variables globales**

**Variables globales**

**ERROR_MESSAGE** : message d’erreur généra par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants AmazonMysql, notamment avec les composants tAmazonMysqlCommit et tAmazonMysqlRollback.</th>
</tr>
</thead>
</table>

**Scénario associé**

Pour un scénario utilisant ce composant, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620
tAmazonMysqlInput

Ce composant lit une base de données et en extrait des champs à l’aide de requêtes.

Le tAmazonMysqlInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux Row > Main.

Propriétés du tAmazonMysqlInput Standard

Ces propriétés sont utilisées pour configurer le tAmazonMysqlInput s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonMysqlInput Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>DB Version</strong></td>
<td>MySQL 5 est disponible.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Remarque :</td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existante entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans</td>
</tr>
</tbody>
</table>
la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Table Name</strong></td>
<td>Nom de la table à lire.</td>
</tr>
<tr>
<td><strong>Query type et Query</strong></td>
<td>Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs correspond à celui défini dans le schéma.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| **Additional JDBC parameters** | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case Use an existing connection est décochée dans les Basic settings. |

**Remarque :**

Lorsque vous devez traiter des données au format date/heure 0000-00-00 00:00:00 utilisant ce composant, définissez les paramètres comme suit :
<table>
<thead>
<tr>
<th><strong>Enable stream</strong></th>
<th>Cochez cette case pour déterminer avec quelles lignes vous souhaitez travailler. Cette option permet d'améliorer les performances.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trim all the String/Char columns</strong></td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td><strong>Trim column</strong></td>
<td>Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| **Global Variables** | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.  
**QUERY** : requête traitée. Cette variable est une variable *Flow* et retourne une chaîne de caractères.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  
Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

**Utilisation**

| **Règle d’utilisation** | Ce composant couvre toutes les possibilités de requête SQL dans les bases de données MySQL. |
| **Dynamic settings** | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant |
la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Scénarios associés

Pour des scénarios associés, consultez tMysqlInput à la page 2631.
tAmazonMysqlOutput

Ce composant écrit, met à jour, modifie ou supprime les données d’une base de données.

Le tAmazonMysqlOutput exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

**Propriétés du tAmazonMysqlOutput Standard**

Ces propriétés sont utilisées pour configurer le tAmazonMysqlOutput s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonMysqlOutput Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>DB Version</td>
<td>MySQL 5 est disponible.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Remarque :</td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td>1.</td>
<td>Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
</tr>
</tbody>
</table>
## tAmazonMysqlOutput

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>

### Username et Password

Informations d’authentification de l’utilisateur de base de données

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

### Table

Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.

### Action on table

Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :

- **Default**: n’effectuer aucune opération sur la table.
- **Drop and create the table**: supprimer la table puis en créer une nouvelle.
- **Create a table**: créer une table qui n’existe pas encore.
- **Create table if doesn’t exist**: créer la table si nécessaire.
- **Drop a table if exists and create**: supprimer la table si elle existe déjà, puis en créer une nouvelle.
- **Clear a table**: supprimer le contenu de la table.
- **Truncate table**: supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.

### Action on data

Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

- **Insert**: Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
- **Update**: Mettre à jour les entrées existantes.
- **Insert or update**: insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Update or insert**: met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
- **Delete**: Supprimer les entrées correspondantes au flux d’entrée.

2. **Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.**

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.
Insert or update on duplicate key or unique index : Ajouter des entrées si la valeur insérée n'existe pas ou mettre à jour les entrées si la valeur insérée existe déjà et qu'un risque de violation d'une clé unique se présente.

Insert ignore : Ajouter uniquement de nouvelles lignes afin d'empêcher les erreurs de doublons de clés.

Avertissement :
Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations Update et Delete. Pour cela, cliquez sur le bouton [...] à côté du champ Edit Schema et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet Advanced settings pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case Use field options et sélectionnez la case Key in update correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases Key in delete pour les opérations de suppression (Delete).

Schema et Edit schema
Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Die on error
Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects.
## Advanced settings

| **Additional JDBC parameters** | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.  

### Remarque :  
Vous pouvez appuyer sur Ctrl+Espace afin d'accéder à une liste de variables globales prédéfinies. |
|---|
| **Extend Insert** | Cochez cette case pour insérer un ensemble de lignes définies au lieu d’insérer les lignes une par une. Cette option permet un important gain de performance.  

*Number of rows per insert* : saisissez le nombre de lignes à insérer en un bloc. Notez que si vous sélectionnez un nombre important de lignes, cela peut augmenter la quantité de mémoire utilisée et donc diminuer les performances.  

### Remarque :  
Cette option n’est pas compatible avec le lien **Reject**. Vous devez donc la décocher si vous utilisez un lien **Reject** en sortie du composant.  

### Avertissement :  
Si vous utilisez ce composant avec un **tMysqlLastInsertID**, soyez sûr de décocher la case **Extend Insert** dans la vue **Advanced settings**. **Extend Insert** permet le chargement par lots, cependant, si la case est cochée, seul l’ID de la dernière ligne du dernier lot sera retourné. |
| **Use Batch** | Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.  

### Remarque :  
Cette case est disponible uniquement si vous avez choisi l’option **Insert, Update, Single Insert Query** ou **Delete** dans le champ **Action on data**. |
| **Batch Size** | Spécifiez le nombre d’enregistrements à traiter dans chaque lot.  
Ce champ est disponible uniquement lorsque la case **Use batch mode** est cochée. |
| **Commit every** | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution. |
| **Additional Columns** | Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des act |
ions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.

| Name | Saisissez le nom de la colonne à modifier ou à insérer. |
| SQL expression | Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes. |
| Position | Sélectionnez Before, Replace ou After, en fonction de l’action à effectuer sur la colonne de référence. |
| Reference column | Saisissez une colonne de référence que le composant tAmazonMysqlOutput peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier. |

Use field options

Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.

Use Hint Options

Cochez cette case pour activer la zone de configuration des indicateurs (ou Hints) permettant d’optimiser l’exécution d’une requête. Dans cette zone, les paramètres sont :
- HINT : spécifiez l’indicateur dont vous avez besoin, en utilisant la syntaxe /*+ */
- POSITION : spécifiez la place de l’indicateur dans une instruction SQL.
- SQL STMT : sélectionnez l’instruction SQL que vous souhaitez utiliser.

Enable debug mode

Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.

Use duplicate key update mode insert

Met à jour les valeurs des colonnes spécifiées en cas de doublon de la clé primaire.

**Column** : Saisissez entre guillemets le nom de la colonne à mettre à jour.

**Value** : Saisissez l’opération que vous souhaitez effectuer sur la colonne.

**Remarque** :
Pour utiliser cette option, vous devez préalablement sélectionner le mode **Insert** dans la liste **Action on data** de la vue **Basic settings**.

Use StatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>Nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong></td>
<td>Nombre de lignes mises à jour. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong></td>
<td>Nombre de lignes insérées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong></td>
<td>Nombre de lignes supprimées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_REJECTED</strong></td>
<td>Nombre de lignes rejetées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>QUERY</strong></td>
<td>Requête traitée. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option. Une variable <em>Flow</em> fonctionne durant l’exécution d’un composant. Une variable <em>After</em> fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <em>Ctrl+Espace</em> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

Utilisation

Règle d’utilisation

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL.

Ce composant doit être utilisé en tant que composant de sortie. Il permet de faire des actions sur une table ou les données d’une table d’une base de données MySQL. Il permet aussi de créer un flux de rejet avec un lien *Row > Reject* filtrant les données en erreur.

Pour un exemple d’utilisation du *tAmazonMysqlOutput*, consultez Scénario : Récupérer les données erronées à l’aide d’un lien *Reject* à la page 2675.

Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant...
la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénarios associés

Pour des scénarios associés, consultez tMysqlSCD à la page 2712.
tAmazonMysqlRollback

Ce composant annule la transaction dans une base de données connectée et évite le commit de transaction involontaire.

Propriétés du tAmazonMysqlRollback Standard

Ces propriétés sont utilisées pour configurer le tAmazonMysqlRollback s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonMysqlRollback Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tAmazonMysqlConnection dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette |
liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec d’autres composants AmazonMySQL, notamment les composants tAmazonMysqlConnection et tAmazonMysqlCommit.</th>
</tr>
</thead>
</table>

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend. |

### Scénario associé

Pour un scénario associé, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623.
tAmazonMysqlRow

Ce composant exécute des requêtes SQL déclarées sur la base de données spécifiée. Selon la nature de la requête et de la base de données, tAmazonMysqlRow agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes. tAmazonMysqlRow est le composant spécifique à ce type de base de données. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tAmazonMysqlRow Standard

Ces propriétés sont utilisées pour configurer le tAmazonMysqlRow s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonMysqlRow Standard appartient aux familles Cloud et Databases. Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>DB Version</td>
<td>MySQL 5 est disponible.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <strong>Guide utilisateur du Studio Talend</strong>.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le <strong>Guide utilisateur du Studio Talend</strong>. Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets. Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Nom de la table à traiter.</td>
</tr>
<tr>
<td>Query type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.</td>
<td></td>
</tr>
</tbody>
</table>
Repository : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.

**Guess Query**

Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.

**Query**

Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

**Advanced settings**

**Additional JDBC parameters**

Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.

**Propagate QUERY’s recordset**

Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.

*i* **Remarque** :

Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type **Object**. Ce composant est généralement suivi du tParseRecordSet.

**Use PreparedStatement**

Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des '?' dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**.

**Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL.

**Parameter Type** : Saisissez le type du paramètre.

**Parameter Value** : Saisissez la valeur du paramètre.

*i* **Remarque** :

Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

**Commit every**

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit
la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.  
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.  
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.  
Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  
La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.  
Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir |
Scénario associé

Pour un scénario associé, consultez :

• Scénario : Combiner deux flux pour une sortie sélective à la page 2706
tAmazonOracleClose

Ce composant ferme la connexion à une base de données connectée.

Propriétés du tAmazonOracleClose Standard

Ces propriétés sont utilisées pour configurer le tAmazonOracleClose s'exécutant dans le framework de Jobs Standard.

Le composant tAmazonOracleClose Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant tAmazonOracleConnection dans la liste.</td>
</tr>
</tbody>
</table>

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

- Une variable *Flow* fonctionne durant l'exécution d'un composant. Une variable *After* fonctionne après l'exécution d'un composant.

- Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec d’autres composants AmazonOracle, notamment tAmazonOracleConnection et tAmazonOracleCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Scénario associé

Ce composant est étroitement lié aux composants tAmazonOracleConnection et tAmazonOracleRollback. Il est généralement utilisé avec un composant tAmazonOracleConnection car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant tAmazonOracleClose, consultez tMysqlConnection à la page 2618.
tAmazonOracleCommit

Ce composant commite en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes, et permet un gain de performance, en utilisant une connexion unique.

Le composant tAmazonOracleCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tAmazonOracleCommit Standard

Ces propriétés sont utilisées pour configurer le tAmazonOracleCommit s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonOracleCommit Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tAmazonOracleConnection dans la liste.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Avertissement :
Si vous utilisez un lien de type Row > Main pour relier le tOracleCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
### Global Variables


### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec des composants AmazonOracle et notamment tAmazonOracleConnection et tAmazonOracleRollback. |
| Dynamic settings | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend. |
Scénario associé

Pour un scénario associé au composant tAmazonOracleCommit, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tAmazonOracleConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Le tAmazonOracleConnection ouvre une connexion vers une base de données afin d’effectuer une transaction.

Propriétés du tAmazonOracleConnection Standard

Ces propriétés sont utilisées pour configurer le tAmazonOracleConnection s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonOracleConnection Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Connection type</td>
<td>Liste déroulante des pilotes disponibles :</td>
</tr>
<tr>
<td>Oracle SID</td>
<td>Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Oracle 11-5 est disponible.</td>
</tr>
<tr>
<td>Use tns file</td>
<td>Cochez cette case pour utiliser les métadonnées d’un contexte contenu dans un fichier tns.</td>
</tr>
</tbody>
</table>

Remarque :

Un même fichier Tns peut contenir plusieurs contextes.
**TNS File** : Renseignez manuellement le chemin d'accès ou cliquez sur le bouton [...] pour parcourir vos dossiers jusqu'au fichier Tns à utiliser.

**Select a DB Connection in Tns File** : Cliquez sur le bouton [...] pour afficher tous les contextes du fichier Tns et sélectionner celui à utiliser.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma</td>
</tr>
</tbody>
</table>

**Username et Password**
Informations d'authentification de l'utilisateur de base de données.
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Additional JDBC parameters**
Spécifiez des informations supplémentaires de connexion à la base de données créée.

> **Remarque** :
Dans ce champ, vous pouvez paramétrer le type d'encodage.

**Use or register a shared DB Connection**
Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s'affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l'exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.

**Advanced settings**

**Auto Commit**
Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et
les modifications doivent être commitées de manière explicite à l'aide du composant correspondant de commit.

Notez que la fonctionnalité d'auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commit pas jusqu'à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d'espace pour gérer vos transactions dans un Job, il est recommandé d'utiliser un composant Commit.

### Variables globales


### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec des composants AmazonOracle, notamment les composants tAmazonOracleCommit et tAmazonOracleRollback. |

### Scénario associé

Pour un scénario associé au composant tAmazonOracleConnection, consultez tMysqlConnection à la page 2618.
tAmazonOracleInput

Ce composant lit une base de données et en extrait des champs à l’aide de requêtes.
tAmazonOracleInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux Row > Main.

Propriétés du tAmazonOracleInput Standard

Ces propriétés sont utilisées pour configurer le tAmazonOracleInput s’exécutant dans le framework de Jobs Standard.
Le composant tAmazonOracleInput Standard appartient aux familles Cloud et Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionne le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Connection type</td>
<td>Liste déroulante des pilotes disponibles : Oracle SID pour identifier exclusivement une base de données spécifique sur un système.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version d’Oracle que vous utilisez.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une...
connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Oracle schema</td>
<td>Nom du schéma Oracle.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur de base de données. 
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository |
| Table name   | Nom de la table de base de données. |
| Query type et Query | Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma. |

**Advanced settings**

<p>| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |</p>
<table>
<thead>
<tr>
<th><strong>Use cursor</strong></th>
<th>Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d’optimiser les performances.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trim all the String/Char columns</strong></td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td><strong>Trim column</strong></td>
<td>Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
</tbody>
</table>

**Global Variables**

| **Global Variables** | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.  
**QUERY** : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

<table>
<thead>
<tr>
<th><strong>Règle d’utilisation</strong></th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données AmazonOracle.</th>
</tr>
</thead>
</table>
| **Dynamic settings** | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre |
dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Limitation


Scénarios associés

Pour des scénarios associés, consultez :

- Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520.
**tAmazonOracleOutput**

Ce composant écrit, met à jour, modifie ou supprime les données d’une base de données.

Le tAmazonOracleOutput exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

**Propriétés du tAmazonOracleOutput Standard**

Ces propriétés sont utilisées pour configurer le tAmazonOracleOutput s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonOracleOutput Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

| **Connection type** | Liste déroulante des pilotes disponibles :
| Oracle SID | Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système. |
| **DB Version** | Sélectionnez la version d'Oracle que vous utilisez. |
| **Host** | Adresse IP du serveur de base de données. |
| **Port** | Numéro du port d'écoute du serveur de base de données. |
| **Database** | Nom de la base de données. |
| **Username et Password** | Informations d'authentification de l'utilisateur de la base de données.
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Oracle schema** | Nom du schéma Oracle. |
| **Table** | Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois. |
| **Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :
| None | n'effectuer aucune opération de table. |
| Drop and create the table | supprimer la table puis en créer une nouvelle. |
| Create a table | créer une table qui n'existe pas encore. |
| Create table if doesn’t exist | créer la table si nécessaire. |
| Drop a table if exists and create | supprimer la table si elle existe déjà, puis en créer une nouvelle. |
| Clear a table | supprimer le contenu de la table. |

**Avertissement :**
Si vous cochez la case *Use an existing connection* et sélectionnez une option différente de *None* dans la liste *Action on table*, une instruction de commit est générée automatiquement avant l'opération de mise à jour/insertion/suppression de données.

| **Action on data** | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : |
**Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.

**Update** : Mettre à jour les entrées existantes.

**Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.

**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

**Delete** : Supprimer les entrées correspondant au flux d’entrée.

**Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations Update et Delete. Pour cela, cliquez sur le bouton [...] à côté du champ Edit Schema et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet Advanced settings pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case Use field options et sélectionnez la case Key in update correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases Key in delete pour les opérations de suppression (Delete).

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository.

**Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en
erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects.

**Advanced settings**

| **Additional JDBC parameters** | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case *Use an existing connection* est décochée dans les *Basic settings*.

**Remarque :** Vous pouvez appuyer sur Ctrl+Espace afin d'accéder à une liste de variables globales prédéfinies.

| **Override any existing NLS_LANG environment variable** | Cochez cette case pour écraser les variables d'environnement NLS_LANG déjà définies.

| **Commit every** | Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant.

| **Additional Columns** | Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.

**Name** : Saisissez le nom de la colonne à modifier ou à insérer.

**SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.

**Position** : Sélectionnez *Before*, *Replace* ou *After*, en fonction de l'action à effectuer sur la colonne de référence.

**Reference column** : Saisissez une colonne de référence que le composant *tOracleOutput* peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

| **Use field options** | Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.

| **Use Hint Options** | Cochez cette case afin d'activer la configuration de la zone des Hints, vous permettant ainsi d'optimiser l'exécution d'une requête. Dans cette zone, les paramètres sont :

- **HINT** : spécifiez l'hint dont vous avez besoin, à l'aide de la syntaxe /*+ */.
- **POSITION** : spécifiez où mettre l’hint dans une instruction SQL.
- **SQL STMT** : sélectionnez l’instruction SQL que vous souhaitez utiliser.

<table>
<thead>
<tr>
<th>Convert columns and table to uppercase</th>
<th>Cochez cette case pour passer les noms de colonnes et de tables en majuscule.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable debug mode</strong></td>
<td>Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.</td>
</tr>
<tr>
<td><strong>Use Batch</strong></td>
<td>Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.</td>
</tr>
<tr>
<td><strong>Batch Size</strong></td>
<td>Spécifiez le nombre d’enregistrements à traiter dans chaque lot. Ce champ est disponible uniquement lorsque la case <strong>Use batch mode</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Support null in “SQL WHERE” statement</strong></td>
<td>Cochez cette case pour prendre en compte les valeurs Null lors d’une instruction WHERE.</td>
</tr>
</tbody>
</table>

**Global Variables**

| Global Variables | 
|------------------|---
| **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. |
| **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. |
| **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. |
| **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. |
| **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. |
| **QUERY** : requête traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. |
| **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. |

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Ce composant doit être utilisé en tant que composant de sortie. Il permet de faire des actions sur une table ou les données d’une table d’une base de données AmazonOracle. Il permet aussi de créer un flux de rejet avec un lien Row &gt; Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Dynamic settings</th>
<th>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le <em>Guide utilisateur du Studio Talend</em>.</th>
</tr>
</thead>
</table>

Scénarios associés

Pour des scénarios associés au composant tAmazonOracleOutput, consultez :

• tMysqlOutput : dans Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667.
tAmazonOracleRollback

Ce composant annule la transaction dans une base de données connectée et évite le commit de transaction involontaire.

Propriétés du tAmazonOracleRollback Standard

Ces propriétés sont utilisées pour configurer le tAmazonOracleRollback s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonOracleRollback Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant <em>AmazonOracleConnection</em> dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décrochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette |
liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est généralement utilisé avec d'autres composants AmazonOracle, notamment les composants tAmazonOracleConnection et tAmazonOracleCommit.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

Lorsqu'un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend. |

### Scénario associé

Pour un scénario associé au composant tAmazonOracleRollback, consultez tMysqlRollback à la page 2694.
tAmazonOracleRow

Ce composant exécute des requêtes SQL déclarées sur la base de données spécifiée.
Selon la nature de la requête et de la base de données, tAmazonOracleRow agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes. tAmazonOracleRow est le composant spécifique à ce type de base de données. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tAmazonOracleRow Standard

Ces propriétés sont utilisées pour configurer le tAmazonOracleRow s'exécutant dans le framework de Jobs Standard.
Le composant tAmazonOracleRow Standard appartient aux familles Cloud et Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
228

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Liste déroulante des pilotes disponibles.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository |
| Query type      | Peut être Built-in ou Repository |
| Query           | Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma. |
| Use NB_LINE_    | Cette option vous permet d’alimenter la variable afin de passer au sous-job ou au composant suivant le nombre de lignes mises à jour/supprimées/insérées. Ce champ s’applique uniquement si la requête saisie dans |
le champ **Query** est une requête INSERT, UPDATE ou DELETE.
- **NONE** : n'alimente pas la variable.
- **INSERTED** : alimente la variable avec le nombre de lignes insérées.
- **UPDATED** : alimente la variable avec le nombre de lignes mises à jour.
- **DELETED** : alimente la variable avec le nombre de lignes supprimées.

**Die on error**
Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

**Propagate QUERY’s recordset**
Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.

**Remarque :**
Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type **Object**. Ce composant est généralement suivi du **tParseRecordSet**.

**Use PreparedStatement**
Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des ‘?’ dans l’instruction SQL définie dans le champ **Query** de l’onglet **Basic settings**.

- **Parameter Index** : Saisissez la position du paramètre dans l’instruction SQL.
- **Parameter Type** : Saisissez le type du paramètre.
- **Parameter Value** : Saisissez la valeur du paramètre.

**Remarque :**
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

**Commit every**
Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.
### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINEUPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINEINSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINEDELETED : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions.
Scénarios associés

Pour des scénarios associés, consultez :

- Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
- tDBSQLRow: Procédure.
- tMySQLRow: Scénario : Supprimer et re-générer un index de table MySQL à la page 2700.
tAmazonRedshiftManage

Ce composant gère des clusters et snapshots Amazon Redshift.

Le tAmazonRedshiftManage gère le travail de création d’un nouveau cluster Amazon Redshift, en créant un snapshot d’un cluster Amazon Redshift, redimensionnant un cluster Amazon Redshift existant et supprimant un cluster ou un snapshot existant.

Propriétés du tAmazonRedshiftManage Standard

Ces propriétés sont utilisées pour configurer le tAmazonRedshiftManage s’exécutant dans le framework de Jobs Standard.

Le composant tAmazonRedshiftManage Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Basic settings

| Access Key et Secret Key | Spécifiez les clés d’accès (l’ID de la clé d’accès dans le champ Access Key et la clé secrète d’accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés d’accès d’AWS, consultez Clés d’accès (ID de clé d’accès et clé d’accès secrète).

Pour saisir la clé secrète, cliquez sur le bouton […] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
</tbody>
</table>
| Assume role | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.

• **Role ARN** : nom Amazon Resource Name (ARN) du rôle.

• **Role session name** : identifiant de la session du rôle.

• **Session duration (minutes)** : durée (en minutes) pour laquelle est active la session du rôle.

Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais). |
| **Action** | Sélectionnez dans la liste une action à effectuer :
• **Create cluster** : crée un cluster Amazon Redshift.
• **Delete cluster** : supprime un cluster Amazon Redshift précédemment fourni.
• **Resize cluster** : redimensionne un cluster Amazon Redshift existant.
• **Restore from snapshot** : crée un nouveau cluster Amazon Redshift depuis un snapshot.
• **Delete snapshot** : supprime le snapshot manuel spécifié.

| **Region** | Spécifiez la région AWS en sélectionnant dans la liste un nom de région ou saisissez un nom de région entre guillemets doubles (par exemple `us-east-1`). Pour plus d’informations concernant comment spécifier la région AWS, consultez Choix d’une région AWS.

| **Create snapshot** | Cochez cette case pour créer un snapshot final du cluster Amazon Redshift avant sa suppression.
Cette case est disponible uniquement lorsque l’option **Delete cluster** est sélectionnée dans la liste **Action**.

| **Snapshot id** | Saisissez l’identifiant du snapshot.
Ce champ est disponible lorsque :
• l’option **Delete cluster** est sélectionnée dans la liste **Action** et que la case **Create snapshot** est cochée.
• l’option **Restore from snapshot** ou **Delete snapshot** est sélectionnée dans la liste **Action**.

| **Cluster id** | Saisissez l’ID du cluster.
Ce champ est disponible lorsque l’option **Create cluster, Delete cluster, Resize cluster** ou **Restore from snapshot** est sélectionnée dans la liste **Action**.

| **Database** | Saisissez le nom de la première base de données à créer lorsque le cluster est créé.
Ce champ est disponible uniquement lorsque l’option **Create cluster** est sélectionnée dans la liste **Action**.

| **Port** | Saisissez le numéro du port sur lequel le cluster accepte les connexions.
Ce champ est disponible lorsque l’option **Create cluster** ou **Restore from snapshot** est sélectionnée dans la liste **Action**.

| **Master username et Master password** | Nom d’utilisateur et mot de passe associé au compte de l’utilisateur maître pour le cluster à créer.
Pour saisir le mot de passe, cliquez sur le bouton `[...]` à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Les deux champs sont disponibles uniquement lorsque l’option <strong>Create cluster</strong> est sélectionnée dans la liste <strong>Action</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Node type</strong></td>
<td>Sélectionnez le type de nœud pour le cluster. Cette liste est disponible uniquement lorsque l’option <strong>Create cluster, Resize cluster ou Restore from snapshot</strong> est sélectionnée dans la liste <strong>Action</strong>.</td>
<td></td>
</tr>
<tr>
<td><strong>Node count</strong></td>
<td>Saisissez le nombre de nœuds de calcul dans le cluster. Ce champ est disponible uniquement lorsque l’option <strong>Create cluster ou Resize cluster</strong> est sélectionnée dans la liste <strong>Action</strong>.</td>
<td></td>
</tr>
</tbody>
</table>

**Advanced settings**

| | | Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées. Cette case est disponible uniquement lorsque la case **Assume role** est cochée. |
| **STS Endpoint** | | |
| **Wait for cluster ready** | Cochez cette case afin de laisser votre Job attendre jusqu’à ce que le lancement du cluster soit terminé. Cette case est disponible uniquement lorsque l’option **Create cluster ou Restore from snapshot** est sélectionnée dans la liste **Action**. |
| **Original cluster id of snapshot** | Saisissez le nom du cluster duquel le snapshot source a été créé. Ce champ est disponible uniquement lorsque l’option **Restore from snapshot ou Delete snapshot** est sélectionnée dans la liste **Action**. |
| **Parameter group name** | Saisissez le nom du groupe de paramètres à associer au cluster. Ce champ est disponible uniquement lorsque l’option **Create cluster ou Restore from snapshot** est sélectionnée dans la liste **Action**. |
| **Subnet group name** | Saisissez le nom du groupe du sous-réseau dans lequel vous souhaitez que le cluster soit restauré. Ce champ est disponible uniquement lorsque l’option **Create cluster ou Restore from snapshot** est sélectionnée dans la liste **Action**. |
| **Publicly accessible** | Cochez cette case afin que le cluster soit accessible depuis un réseau public. Cette case est disponible uniquement lorsque l’option **Create cluster ou Restore from snapshot** est sélectionnée dans la liste **Action**. |
### Set public ip address

Cochez cette case et, dans le champ qui s’affiche, saisissez l’adresse IP d’Elastic IP (EIP) pour le cluster.

Cette case est disponible uniquement lorsque la case **Publicly accessible** est cochée.

### Availability zone

Saisissez la zone de disponibilité EC2 (EC2 Availability Zone) dans laquelle vous souhaitez qu’Amazon Redshift fournisse le cluster.

Ce champ est disponible uniquement lorsque l’option **Create cluster** ou **Restore from snapshot** est sélectionnée dans la liste **Action**.

### VPC security group ids

Saisissez les groupes de sécurité du Cloud privé virtuel (Virtual Private Cloud, VPC) à associer au cluster et séparez-les à l’aide d’une virgule, par exemple, *gname1, gname2, gname3*.

Ce champ est disponible lorsque l’option **Create cluster** ou **Restore from snapshot** est sélectionnée dans la liste **Action**.

### tStatCatcher Statistics

Cochez cette case afin de collecter les informations de log au niveau du Job ainsi qu’au niveau des composants.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>CLUSTER_FINAL_ID : ID du cluster. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ENDPOINT : adresse de l’endpoint du cluster. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

### Utilisation

| Règle d’utilisation | Le **tAmazonRedshiftManage** est généralement utilisé en tant que composant standalorne. |
Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tApacheLogInput

Ce composant lit le fichier de log d’accès d’un serveur Apache.

Afin de gérer efficacement un serveur Apache, il est obligatoire lorsqu’il s’agit d’obtenir un retour sur l’activité et les performances du serveur, mais aussi sur les problèmes susceptibles de se produire.

Propriétés du tApacheLogInput Standard

Ces propriétés sont utilisées pour configurer le tApacheLogInput s’exécutant dans le framework de Jobs Standard.

Le composant tApacheLogInput Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Dans le contexte d’utilisation du tApacheLogInput, le schéma est en lecture seule.


File Name

Chemin d’accès et nom du fichier, et/ou variable à traiter.

Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend.

Die on error

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject.
Advanced settings

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en standalone. Il permet de créer un flux de données à l’aide d’un lien Row > Main, ainsi que de créer un flux de rejet avec un lien Row > Reject filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d’utilisation de ces deux liens, consultez Procédure à la page 1152 du composant tFileInputXML. |

Scénario : Lire un fichier de log d’accès Apache

Le scénario suivant crée un Job à deux composants permettant de lire le fichier de log d’accès d’un serveur Apache et d’afficher les données en sortie dans la console de log Run.

Procédure

1. A partir de la Palette, cliquez-déposez les composants tApacheLogInput et tLogRow dans l’éditeur graphique.
2. Cliquez-droit sur le **tApacheLogInput** et sélectionnez une connexion **Row > Main** dans le menu contextuel. Cliquez ensuite sur le **tLogRow**, le lien approprié apparaît entre les deux composants.

3. Dans l’espace de modélisation graphique, sélectionnez le **tApacheLogInput**.

4. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du composant **tApacheLogInput**.

5. Configurez les champs **Property Type** et **Schema** en mode **Built-In**.

6. Cliquez sur le bouton **Edit schema** si vous souhaitez voir les colonnes (en mode read-only).

7. Dans le champ **File Name**, renseignez le chemin d’accès ou parcourez vos dossiers jusqu’au fichier de log d’accès que vous voulez lire.

8. Dans l’éditeur graphique, sélectionnez le composant **tLogRow** et cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**). Pour plus d’informations, consultez **tLogRow** à la page 2105.

9. Appuyez sur **F6** pour exécuter le Job.

Résultats

Les lignes de log du fichier défini sont affichées dans la console.
**tAS400Close**

Ce composant ferme la connexion à une base de données connectée.

**Propriétés du tAS400Close Standard**

Ces propriétés sont utilisées pour configurer le tAS400Close s’exécutant dans le framework de Jobs Standard.

Le composant tAS400Close Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tAS400Connection dans la liste.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tAS400Commit

Ce composant commit une transaction globale en une seule fois au lieu de commiter chaque ligne ou chaque lot de lignes, et permet un gain de performance, en utilisant une connexion unique.

Le composant tAS400Commit valide les données traitées par le Job dans la base de données connectée.

Propriétés du tAS400Commit Standard

Ces propriétés sont utilisées pour configurer le tAS400Commit s’exécutant dans le framework de Jobs Standard.

Le composant tAS400Commit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tAS400Connection dans la liste.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
<tr>
<td></td>
<td><strong>Avertissement :</strong></td>
</tr>
<tr>
<td></td>
<td>Si vous utilisez un lien de type <strong>Row &gt; Main</strong> pour relier le tAS400Commit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case <strong>Close connection</strong> car la connexion sera fermée avant la fin du commit de votre première ligne.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics   | Cochez cette case pour collecter les données de log au niveau du composant. |
## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants AS/400 et notamment avec les composants tAS400Connection et tAS400Rollback.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

## Scénario associé

Pour un scénario associé au composant tAS400Commit, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tAS400Connection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Le tAS400Connection ouvre une connexion vers une base de données afin d’effectuer une transaction.

**Propriétés du tAS400Connection Standard**

Ces propriétés sont utilisées pour configurer le tAS400Connection s’exécutant dans le framework de Jobs Standard.

Le composant tAS400Connection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th><strong>Database</strong></th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>DB Version</strong></td>
<td>Sélectionnez la version d’AS/400 que vous utilisez.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Use or register a shared DB Connection</strong></td>
<td>Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ <strong>Shared</strong>...</td>
</tr>
</tbody>
</table>
DB Connection Name qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options Use dynamic job et Use an independent process to run subjob du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.

### Advanced settings

<table>
<thead>
<tr>
<th>Additional JDBC parameters</th>
<th>Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case Use an existing connection est décochée dans les Basic settings.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Commit</td>
<td>Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit. Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commit pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants AS/400 et notamment avec les composants tAS400Commit et tAS400Rollback.</th>
</tr>
</thead>
</table>

### Scénario associé

Pour plus d’informations relatives au fonctionnement du composant tAS400Connection, consultez tMysqlConnection à la page 2618.
tAS400Input

Ce composant lit une base de données et en extrait des champs à l’aide de requêtes.
tAS400Input exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux Row > Main.

Propriétés du tAS400Input Standard

Ces propriétés sont utilisées pour configurer le tAS400Input s’exécutant dans le framework de Jobs Standard.
Le composant tAS400Input Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

.dw

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.
<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
<td></td>
</tr>
</tbody>
</table>
| Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant.  
Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend. |
| **DB Version** | Sélectionnez la version d’AS/400 que vous utilisez. |
| **Host** | Adresse IP du serveur de base de données. |
| **Port** | Numéro du port d'écoute du serveur. |
| **Database** | Nom de la base de données. |
| **Username et Password** | Informations d'authentification de l'utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. |
**Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th>Query type et Query</th>
<th>Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.</th>
</tr>
</thead>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Additional JDBC parameters</th>
<th>Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case <strong>Use an existing connection</strong> est décochée dans les <strong>Basic settings</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim all the String/Char columns</td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaines de caractères.</td>
</tr>
<tr>
<td>Trim column</td>
<td>Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL.</th>
</tr>
</thead>
</table>
| Dynamic settings | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.  

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez [Scénario : Lire des données dans des bases de données à l’aide de connexions](#). |
Scénario : Gérer des données à l'aide de AS/400

Ce scénario décrit un Job qui écrit et lit les informations utilisateur dans AS/400 et qui les affiche ensuite dans la console.

Ajouter et relier les composants

Procédure
1. Créez un nouveau Job et ajoutez un tFixedFlowInput, un tAS400Output, un tAS400Input et un tLogRow en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le tFixedFlowInput au tAS400Output à l'aide d'un lien Row > Main.
3. Faites la même chose pour relier le tAS400Input au tLogRow.
4. Reliez le tFixedFlowInput au tAS400Input à l'aide d'un lien Trigger > OnSubjobOk.

Configurer les composants

Écrire les données dans AS/400

Procédure
1. Double-cliquez sur le tFixedFlowInput pour ouvrir sa vue Basic settings.
2. Cliquez sur le bouton … à côté de Edit schema et dans la boîte de dialogue [Schema], définissez le schéma en ajoutant trois colonnes : id de type Integer ainsi que name et city de type String.

Cliquez sur OK pour fermer la boîte de dialogue [Schema] et acceptez la propagation proposée par la fenêtre qui s’affiche.

3. Dans la zone Mode, sélectionnez Use Inline Content (delimited file) et saisissez les informations utilisateur suivantes dans le champ Content.

1;George;Bismarck
2;Abraham;Boise
3;Taylor;Nashville
4;William;Jefferson City
5;Alexander;Jackson
6;James;Boise
7;Gerald;Little Rock
8;Tony;Richmond
9;Thomas;Springfield
10;Andre;Nashville
4. Double-cliquez sur le **tAS400Output** pour ouvrir sa vue **Basic settings**.


6. Dans le champ **Table**, spécifiez la table dans laquelle vous souhaitez écrire les données. Dans cet exemple, la table est *doct1018*.

7. Sélectionnez **Drop table if exists and create** dans la liste déroulante **Action on table** et **Insert** dans la liste déroulante **Action on data**.

**Récupérer les données depuis AS/400**

**Procédure**

1. Double-cliquez sur le **tAS400Input** pour ouvrir sa vue **Basic settings**.

3. Cliquez sur le bouton [...] à côté de Edit schema et dans la boîte de dialogue [Schema], définissez le schéma en ajoutant trois colonnes : id de type Integer ainsi que name et city de type String. La structure des données est identique à celle que vous avez définie pour le tFixedFlowInput.

4. Dans le champ Table Name, saisissez le nom de la table dans laquelle écrire les données ou bien parcourez votre système jusqu’à celle-ci. Dans cet exemple, la table est doct1018.

5. Dans le champ Query, saisissez la requête SQL à utiliser pour récupérer les données utilisateur depuis AS/400. Dans cet exemple, la requête est SELECT * FROM doct1018.

6. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings.

7. Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour afficher un résultat plus lisible.

**Enregistrer et exécuter le Job**

**Procédure**

1. Appuyez sur Ctrl + S pour enregistrer le Job.

2. Appuyez sur F6 ou cliquez sur Run dans l’onglet Run pour exécuter le Job.

```
[statistics] connecting to socket on port 3565
[statistics] connected
+--------------------------+
| tLogRow_1                |
| id | name      | city        |
+--------------------------+
| 1  | George    | Bisnerck    |
| 2  | Abraham   | Boise       |
| 3  | Taylor    | Nashville   |
| 4  | William   | Jefferson City |
| 5  | Alexander | Jackson     |
| 6  | James     | Boise       |
| 7  | Gerald    | Little Rock |
| 8  | Tony      | Richmond    |
| 9  | Thomas    | Springfield |
| 10 | Andre     | Nashville   |
+--------------------------+
[statistics] disconnected
```

Comme affiché ci-dessus, les informations utilisateur sont écrites dans AS/400, puis y sont récupérées et affichées dans la console.

**Scénarios associés**

Pour un scénario associé utilisant d’autres bases de données, consultez : dans :
Consultez également Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520 du composant tContextLoad.
**tAS400LastInsertId**

Ce composant récupère les clés primaires des entrées dernièrement ajoutées à une table AS/400. Le tAS400LastInsertId affiche les derniers ID ajoutés à une table à partir d'une connexion AS/400 spécifiée.

**Propriétés du tAS400LastInsertId Standard**

Ces propriétés sont utilisées pour configurer le tAS400LastInsertId s'exécutant dans le framework de Jobs Standard.

Le composant tAS400LastInsertId Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications.
modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th>Component list</th>
<th>Sélectionnez le composant <strong>tAS400Connection</strong> dans la liste s’il y a plus d’une connexion dans votre Job.</th>
</tr>
</thead>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant intermédiaire.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton `[+]` pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.


**Scénario associé**

Pour un scénario associé, consultez Scénario : Récupérer les ID des dernières entrées ajoutées avec le **tMysqlLastInsertId** à la page 2655 du composant **tMysqlLastInsertId**.
tAS400Output

Ce composant écrit, met à jour, modifie ou supprime les données d’une base de données.
Le tAS400Output exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Propriétés du tAS400Output Standard

Ces propriétés sont utilisées pour configurer le tAS400Output s’exécutant dans le framework de Jobs Standard.
Le composant tAS400Output Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><img src="image" alt="Configuration de connexion à la base de données" /></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version d’AS/400 que vous utilisez.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
</tbody>
</table>

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

<table>
<thead>
<tr>
<th>Table</th>
<th>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois.</th>
</tr>
</thead>
</table>
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :

  - **None** : n’effectuer aucune opération de table.
  - **Drop and create the table** : supprimer la table puis en créer une nouvelle.
  - **Create a table** : créer une table qui n’existe pas encore.
  - **Create table if doesn’t exist** : créer la table si nécessaire.
  - **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.
  - **Clear a table** : supprimer le contenu de la table.

| Action on data | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

  - **Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
  - **Update** : Mettre à jour les entrées existantes.
  - **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. |
**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

**Delete** : Supprimer les entrées correspondantes au flux d’entrée.

**Avertissement** :
Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update et Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

---

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans
le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

| **Die on error** | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects. |

**Advanced settings**

| **Use commit control** | Cochez cette case pour accéder au champ Commit every et définir les opérations de commit.  
Commit every : nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de transaction (cependant pas de rollback) et surtout une meilleure performance des exécutions. |

| **Additional JDBC parameters** | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case Use an existing connection est décochée dans les Basic settings.  
Remarque : Vous pouvez appuyer sur Ctrl+Espace afin d’accéder à une liste de variables globales prédéfinies. |

| **Additional Columns** | Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétreatment particulier.  
Name : Saisissez le nom de la colonne à modifier ou à insérer.  
SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.  
Position : Sélectionnez Before, Replace ou After, en fonction de l’action à effectuer sur la colonne de référence.  
Reference column : Saisissez une colonne de référence que le composant tAS400Output peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier. |

<p>| <strong>Use field options</strong> | Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données. |</p>
<table>
<thead>
<tr>
<th>Tableau</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable debug mode</strong></td>
<td>Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.</td>
</tr>
<tr>
<td><strong>Use Batch</strong></td>
<td>Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.</td>
</tr>
<tr>
<td><strong>Batch Size</strong></td>
<td>Spécifiez le nombre d’enregistrements à traiter dans chaque lot. Ce champ est disponible uniquement lorsque la case <strong>Use batch mode</strong> est cochée.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Utilisation**

**Règle d’utilisation**

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données AS/400. Il permet aussi de créer un flux de rejet avec un lien **Row > Reject** filtrant les données en erreur. Pour un exemple d’utilisation, consultez **Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput**.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.
Scénarios associés

Pour un scénario associé, consultez Scénario : Gérer des données à l'aide de AS/400 à la page 249.

Pour des scénarios associés utilisant d'autres bases de données, consultez :

• Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667.
tAS400Rollback

Ce composant annule la transaction dans une base de données connectée et évite le commit de transaction involontaire.

Propriétés du tAS400Rollback Standard

Ces propriétés sont utilisées pour configurer le tAS400Rollback s'exécutant dans le framework de Jobs Standard.

Le composant tAS400Rollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant de connexion tAS400Connection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé en association avec des composants AS/400, notamment avec le tAS400Connection et le tAS400Commit. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes |
bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

Lorsqu'un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Scénario associé

Pour un scénario associé au tAS400Rollback, consultez Scénario : Annuler l'insertion de données dans des tables mère/fille à la page 2623 du composant tMySQLRollback.
tAS400Row

Ce composant exécute des requêtes SQL déclarées sur la base de données spécifiée.

Selon la nature de la requête et de la base de données, tAS400Row agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes. tAS400Row est le composant spécifique à ce type de base de données. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tAS400Row Standard

Ces propriétés sont utilisées pour configurer le tAS400Row s’exécutant dans le framework de Jobs Standard.

Le composant tAS400Row Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux
<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être <em>Built-in</em> ou <em>Repository</em>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version d’AS/400 que vous utilisez.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| Username et Password     | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ *Password*, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Schema et Edit Schema    | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
**Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• **View schema** : sélectionnez cette option afin de voir le schéma.  
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la... |
| **Table Name** | Nom de la table à traiter. |
| **Query type** | La requête peut être **Built-in** ou distante dans le **Repository** |
| **Query** | Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma. |
| **Die on error** | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**. |

**Advanced settings**

| **Additional JDBC Parameters** | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. |
| **Propagate QUERY’s recordset** | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. | **Remarque** : Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type **Object**. Ce composant est généralement suivi du **tParseRecordSet**. |
| **Use PreparedStatement** | Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des '?' dans l’instruction SQL définie dans le champ **Query** de l’onglet **Basic settings**. |
| **Parameter Index** | Saisissez la position du paramètre dans l’instruction SQL. |
| **Parameter Type** | Saisissez le type du paramètre. |
| **Parameter Value** | Saisissez la valeur du paramètre. |
### Remarque :
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

<table>
<thead>
<tr>
<th>Commit every</th>
<th>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log lors du traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Utilisation

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL. |
| Dynamic settings | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table Dynamic settings est disponible uniquement lorsqu’une case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.  


### Scénarios associés

Pour un scénario associé, consultez :

- Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
- Procédure du composant tDBSQLRow.
• Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du composant tMysqlRow.
tAssert

Ce composant génère l’assertion booléenne relative à l’analyse du Job et envoie le message de statut du Job au tAssertCatcher.

Le statut peut être :
• **Ok** : l’exécution du Job a réussi.
• **Fail** : l’exécution a échoué.

Les résultats du Job testé ne correspondent pas à ce qui était attendu, ou une erreur est survenue lors de l’exécution.

Le composant tAssert est étroitement lié au tAssertCatcher pour tester une exécution de Job. Il génère une assertion booléenne à partir d’une instruction conditionnelle, dans le but d’alimenter le statut d’exécution présenté par le tAssertCatcher.

Propriétés du tAssert Standard

Ces propriétés sont utilisées pour configurer le tAssert s’exécutant dans le framework de Jobs Standard.

Le composant tAssert Standard appartient à la famille Logs & Errors.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Description</th>
<th>Saisissez votre message descriptif afin d’identifier l’assertion du tAssert.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression</td>
<td>Saisissez l’instruction conditionnelle sur laquelle baser votre assertion.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métdonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace |

269
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant suit l’action à laquelle est liée directement la condition assertive. Il peut être utilisé comme composant intermédiaire ou final du Job principal, ou bien en tant que composant de début, intermédiaire ou de fin du Job secondaire.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>L’évaluation du tAssert est capturée par le tAssertCatcher.</td>
</tr>
</tbody>
</table>

Scénario 1: Obtenir le statut des commandes d’un produit (quotidiennement) par rapport à un nombre fixe

Ce scénario permet d’insérer dans une table de base de données les informations liées à des commandes, ainsi que d’obtenir chaque jour le statut de ces commandes en utilisant le composant tAssert pour comparer les commandes à un nombre fixe et tAssertCatcher pour indiquer les résultats. Ici, Ok est retourné lorsque le nombre de commandes est supérieur à 20 et Failed est retourné s’il est inférieur à 20.

En pratique, ce Job peut être planifié pour s’exécuter tous les jours et fournir un rapport quotidien des commandes, et vous pouvez aussi remplacer les composants tFixedFlowInput et tLogRow par des composants d’entrée et de sortie dans les familles Database et File.

Déposer et relier les composants

Procédure

1. Déposez les composants tFixedFlowInput, tMysqlOutput, tAssert, tAssertCatcher, et tLogRow de la Palette dans l’espace de modélisation graphique.
2. Renommez tFixedFlowInput en orders, tAssert en orders >= 20, tAssertCatcher en catch comparison result et tLogRow en ok or failed.
3. Reliez le composant tFixedFlowInput à tMysqlOutput à l’aide d’un lien Row > Main.
4. Reliez le composant tFixedFlowInput à tAssert à l’aide d’un lien Trigger > On Subjob Ok.
5. Reliez le composant tAssertCatcher à tLogRow à l’aide d’un lien Row > Main.
Configurer les composants

Procédure

1. Double-cliquez sur le tFixedFlowInput pour ouvrir sa vue Basic settings.

Sélectionnez Use Inline Content (delimited file) dans la zone Mode.

Dans le champ Content, saisissez les données à passer en base MySQL, par exemple:

```
AS2152;Washington Berry Juice;2013-02-19 11:14:15;3.6
AS2152;Washington Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washington Berry Juice;2013-02-19 13:14:15;3.6
AS2152;Washington Berry Juice;2013-02-19 14:14:15;3.6
AS2152;Washington Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washington Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washington Berry Juice;2013-02-19 12:14:15;3.6
AS2152;Washington Berry Juice;2013-02-19 12:14:15;3.6
```
Notez que ces commandes, inférieures à 20, sont données à titre d'exemple du fonctionnement du tAssert.

2. Cliquez sur le bouton Edit schema pour ouvrir l'éditeur de schéma.

3. Cliquez sur le bouton [+] pour ajouter quatre colonnes, nommées respectivement product_id, product_name, date et price, de type String, Date, Float. Cliquez sur OK pour valider la configuration et fermer l'éditeur.

4. Double-cliquez sur le tMysqlOutput pour ouvrir sa vue Basic settings.

5. Dans les champs Host, Port, Database, Username et Password, saisissez les identifiants et les informations de connexion à la base.

6. Dans le champ Table, saisissez le nom de la table, order par exemple.

7. Dans la liste Action on table, sélectionnez l'option Drop table if exists and create.

8. Dans la liste Action on data, sélectionnez l'option Insert.
9. Double-cliquez sur le tAssert pour ouvrir sa vue Basic settings.

10. Dans le champ description, saisissez les informations descriptives de l’objectif du tAssert dans ce scénario.

11. Dans le champ expression, saisissez l’expression permettant de comparer les données au nombre fixe:

\[((\text{Integer})\text{globalMap.get("tMysqlOutput_1_NB_LINE_INSERTED")]\geq 20\]

12. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings.

13. Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour afficher le résultat sous forme de tableau.

Exécuter le Job

Procédure

1. Appuyez sur Ctrl+S pour sauvegarder votre Job.

Comme montré dans la capture d’écran, le statut des commandes indique Failed puisque le nombre de celles-ci est inférieur à 20.

Scénario 2 : Paramétrer une condition assertive pour l’exécution d’un Job

Ce scénario décrit comment paramétrer une condition assertive dans le composant tAssert afin d’évaluer si l’exécution du Job a réussi ou non. De plus, vous pouvez également trouver comment
deux résultats différents d’évaluation s’affichent et comment les lire. Le scénario utilise un **tAssert**, mais aussi :

- un **tFileInputDelimited** et un **tFileOutputDelimited**. Les deux composants constituent le Job principal, duquel le statut d’exécution est évalué. Pour plus d’informations sur ces deux composants, consultez **tFileInputDelimited** à la page 1067 et **tFileOutputDelimited** à la page 1169.

- un **tFileCompare**. Il réalise la comparaison entre le fichier de sortie du Job principal et un fichier de référence standard. Le résultat comparatif est évalué par le **tAssert** par rapport à la condition assertive configurée dans ses paramètres. Pour plus d’informations à propos du **tFileCompare**, consultez **tFileCompare** à la page 1035.

- un **tFileCompare**. Il capture l’évaluation générée par le **tAssert**. Pour plus d’informations à propos du composant **tAssertCatcher**, consultez **tAssertCatcher** à la page 279.

- un **tLogRow**. Il vous permet de lire l’évaluation capturée. Pour plus d’informations à propos du **tLogRow**, consultez **tLogRow** à la page 2105.

Afin de créer le Job principal, procédez comme suit :

- Préparez un fichier délimité .csv comme fichier source lu par votre Job principal.

- Éditez deux lignes dans ce fichier. Les modifications n’ont pas d’importance, vous pouvez uniquement simplifier le contenu.

- Nommez-le **source.csv**.

- Dans le **Studio Talend**, créez un nouveau Job, que vous appellerez **JobAssertion**.

- Placez un **tFileInputDelimited** et un **tFileOutputDelimited** dans l’espace de modélisation graphique.

- Connectez-les à l’aide d’un lien **Row Main** afin de créer le Job principal.

- Double-cliquez sur le **tFileInputDelimited** afin d’ouvrir sa vue **Component**.

- Dans le champ **File Name** de la vue **Component**, renseignez le chemin d’accès à votre fichier **source.csv**.
• Sélectionnez dans la liste Property Type le mode Built-In et cliquez sur le bouton [...] de l’option Edit schema afin de définir les données à passer au tFileOutputDelimited. Dans ce scénario, sélectionnez les données présentées dans le fichier source.csv que vous avez créé.

Pour plus d’informations à propos des types de schéma, consultez le Guide utilisateur du Studio Talend.

• Définissez les autres paramètres dans les champs correspondants, selon le fichier source.csv que vous avez créé.

• Double-cliquez sur le tFileOutputDelimited afin d’ouvrir sa vue Component.

• Dans le champ File Name de la vue Component, renseignez le chemin d’accès à votre fichier de sortie, et laissez les autres champs comme définis par défaut.

• Appuyez sur F6 afin d’exécuter le Job principal. Il lit votre fichier source.csv, passe les données au tFileOutputDelimited et écrit un fichier délimité de sortie, out.csv.

Continuez à modifier votre Job, afin de voir comment le tAssert évalue le statut d’exécution du Job principal.

• Renommez le fichier out.csv en reference.csv. Ce fichier est le résultat attendu de l’écriture du Job principal.

• Placez un tFileCompare, un tAssert et un tLogRow dans l’espace de modélisation graphique.

• Connectez-les avec un lien Row Main.

• Connectez le tFileInputDelimited au tFileCompare avec un lien OnSubjobOk.

• Double-cliquez sur le tFileCompare afin d’ouvrir sa vue Component.
• Dans la vue Component, renseignez les chemins d’accès dans les champs File to compare et Reference file correspondants, et laissez les autres champs comme définis par défaut.

Pour plus d’informations à propos du composant tFileCompare, consultez tFileCompare à la page 1035.

• Cliquez ensuite sur le tAssert, puis cliquez sur l’onglet Component, au bas de l’espace de modélisation graphique.

• Dans la vue Component, éditez la condition row2.differ==0, dans le champ expression ainsi que la description de l’assertion.

Dans le champ expression, row2 constitue le flux de données transmis du tFileCompare au tAssert, differ est une colonne du schéma du tFileCompare qui affiche si les fichiers comparés sont identiques. 0 signifie qu’aucune différence n’a été détectée entre le fichier out.csv et le fichier reference.csv par le tFileCompare. Les fichiers comparés sont donc identiques, la condition assertive est remplie, le tAssert conclut que le Job principal a été exécuté avec succès. Sinon, il conclut qu’il a échoué.

Remarque :
La colonne differ fait partie du schéma, en lecture seule, du tFileCompare. Pour plus d’informations sur son schéma, consultez tFileCompare à la page 1035.

• Appuyez sur F6 afin d’exécuter votre Job.
• Vérifiez le résultat retourné dans la vue Run.
La console affiche les résultats de la comparaison du \texttt{tFileCompare} : Files are identical. Mais l'évaluation du \texttt{tAssert} n'est visible nulle part.

Vous avez donc besoin du composant \texttt{tAssertCatcher} pour capturer l'évaluation.

- Placez un \texttt{tAssertCatcher} et un \texttt{tLogRow} dans l'espace de modélisation graphique.
- Connectez-les à l'aide d'un lien \texttt{Row Main}.

- Utilisez la configuration par défaut du \texttt{tAssertCatcher} dans sa vue Component.
• Appuyez sur F6 afin d'exécuter votre Job.

• Vérifiez les résultats présentés dans la vue Run. Vous pouvez voir que les informations sur le statut du Job ont été ajoutées :

  2010-01-29 15:37:33|fAvAzH|TASSERT|JobAssertion|java|tAssert_1|Ok|--| The output file should be identical with the reference file.

La description du JobAssertion dans la console est organisée selon le schéma du tAssertCatcher, lequel comprend des colonnes correspondant au temps d'exécution, à l'identifiant de processus, au nom du projet, au langage du code, l'origine de l'évaluation, le résultat de l'évaluation, les informations détaillées de l'évaluation, et le message descriptif de l'assertion. Pour plus d'informations à propos du schéma du tAssertCatcher, consultez tAssertCatcher à la page 279.

La console indique que le statut d'exécution du Job JobAssertion est Ok. En plus de l'évaluation, vous pouvez toujours voir les informations descriptives du JobAssertion, parmi lesquelles la description que vous avez saisie dans les Basic settings du tAssert.

Effectuez ensuite quelques opérations pour que le Job principal ne puisse générer le fichier attendu. Pour ce faire, procédez comme suit :

• Effacez une ligne de votre fichier reference.csv.

• Appuyez sur F6 pour exécuter votre Job à nouveau.

• Vérifiez les résultats retournés dans la vue Run.

  2010-02-01 19:47:43|GeHJNO|TASSERT|JobAssertion|tAssert_1|Failed|Test logically failed|The output file should be identical with the reference file.

La console indique que le statut d'exécution du Job principal est Failed (a échoué). L'explication détaillée suit peu après, Test logically failed (échec logique du test).

Vous pouvez avoir une idée du statut de votre Job : il a échoué lors de la génération du fichier attendu à cause d'un échec logique. Cet échec logique peut provenir d'une erreur logique lors de la création du Job.

Le statut et ses explications sont présentés respectivement dans les colonnes de statut et de sous-statut du schéma du tAssertCatcher. Pour plus d'informations à propos de cette colonne, consultez tAssertCatcher à la page 279.
**tAssertCatcher**

Ce composant génère un flux de données consolidant les informations de statut d’une exécution de Job, et transfère les données dans les fichiers de sortie définis.

A partir de son schéma prédéfini, le tAssertCatcher retrouve les informations des statuts d’exécution, des exécutions des Jobs et du tAssert.

**Propriétés du tAssertCatcher Standard**

Ces propriétés sont utilisées pour configurer le tAssertCatcher s’exécutant dans le framework de Jobs Standard.

Le composant tAssertCatcher Standard appartient à la famille Logs & Errors.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit schema | Un schéma est une description de lignes, il définit les champs à traiter et à passer au composant suivant. Pour ce composant, le schéma est en lecture seule puisqu’il regroupe les informations de log standard, notamment :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Moment</strong></td>
<td>Date et heure de traitement.</td>
</tr>
<tr>
<td><strong>Pid</strong></td>
<td>Identifiant de processus.</td>
</tr>
<tr>
<td><strong>Project</strong></td>
<td>Projet auquel appartiennent les Jobs analysés.</td>
</tr>
<tr>
<td><strong>Job</strong></td>
<td>Nom du Job analysé.</td>
</tr>
<tr>
<td><strong>Origin</strong></td>
<td>Origine de l'assertion de statut. Elle peut provenir de différents composants tAssert.</td>
</tr>
</tbody>
</table>
| **Status** | Jugements retrouvés à partir du tAssert. Ils peuvent être :
- **Ok** : si l'instruction conditionnelle du tAssert est évaluée comme vraie (true) à l'exécution.
- **Failed** : si l'instruction conditionnelle du tAssert est évaluée comme fausse (false), ou si des erreurs surviennent à l'exécution. Les résultats du Job testé ne correspondent pas à ce qui était attendu, ou une erreur est survenue lors de l’exécution. |
| **Substatus** | Explications détaillées à propos des assertions de statut. Les explications peuvent être :
- **Test logically failed** : le Job analysé ne produit pas le résultat attendu.
- **Execution Error** : une erreur d'exécution est survenue lors de l’exécution. |
<p>| <strong>Description</strong> | La description textuelle saisie dans l’onglet Basic settings du tAssert lorsquand la case Catch tAssert est cochée. |</p>
<table>
<thead>
<tr>
<th>AssertCatcher</th>
<th>est cochée) et/ou le message d’exception capturée (lorsque la case Catch Java Exception est cochée).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exception</strong></td>
<td>l’objet de l’exception retournée par le Job, l’exception originale. Disponible lorsque la case Get original exception est cochée.</td>
</tr>
<tr>
<td><strong>Catch Java Exception</strong></td>
<td>Cette case, une fois cochée, vous permet de capturer les erreurs d’exception Java et d’afficher le message dans la colonne Description (quand la case Get original exception n’est pas cochée) ou dans la colonne Exception (quand la case Get original exception est cochée).</td>
</tr>
<tr>
<td><strong>Get original exception</strong></td>
<td>Cette case, une fois cochée, permet d’afficher l’objet de l’exception originale dans la colonne Exception. Disponible lorsque la case Catch Java Exception est cochée.</td>
</tr>
<tr>
<td><strong>Catch tAssert</strong></td>
<td>Cette case vous permet de capturer les assertions du tAssert.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant est un composant de début de Job secondaire qui retrouve les informations de statut d’exécution à partir de différentes sources. Il génère un |
flux de données afin de transférer les informations au composant suivant.

| Limitation | Ce composant doit être utilisé avec le composant tAssert. |

Scénario associé

Pour un scénario associé, consultez le scénario du tAssert :

- Scénario 2 : Paramétrer une condition assertive pour l’exécution d’un Job à la page 273.
tAzureStorageConnection

Ce composant utilise des informations d'authentification et de protocole afin de créer une connexion au système Microsoft Azure Storage pouvant être réutilisée par d'autres composants Azure Storage.

Propriétés du tAzureStorageConnection Standard

Ces propriétés sont utilisées pour configurer le tAzureStorageConnection s'exécutant dans le framework de Jobs Standard.
Le composant tAzureStorageConnection Standard appartient à la famille Cloud.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• Built-In : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• Repository : les informations de connexion stockées centralement dans le <code>Repository &gt; Metadata</code> seront réutilisées par ce composant. Vous devez cliquer sur le bouton <code>[...]</code> et, dans la boîte de dialogue <code>Repository Content</code>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Account Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l'administrateur du système a donné les droits d'accès appropriés à ce compte de stockage.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Account Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n'importe laquelle peut être utilisée pour accéder au compte.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sélectionnez le protocole de la connexion à créer.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Azure Shared Access Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cochez cette case pour utiliser une signature d'accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d'informations, consultez Utilisation des signatures d'accès partagé (SAP).</td>
</tr>
</tbody>
</table>
|                                  | Dans le champ `Azure Shared Access Signature` qui s'affiche, saisissez votre signature d'accès partagé entre guillemets doubles. Vous pouvez obtenir l'URL de la signature d'accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la
signature. Le format de l’URL est le suivant :
https://<$storagename>.<$service>.core.windows.net/<$sastoken>, où <$storagename> est
le nom du compte de stockage, <$service> le nom
du service autorisé (blob, file, queue ou table) et <$sastoken> est la valeur du jeton de signature
d’accès partagé. Pour plus d’informations, consultez
Constructing the Account SAS URI (en anglais).

Notez que SAS a une période de validité, vous pouvez
configurer l’heure de début de validité de SAS et son
heure d’expiration, après laquelle SAS n’est plus valide
lors de la génération. Vous devez vous assurer que votre
SAS est toujours valide lorsque vous exécutez votre Job.

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de
|                         | traitement du Job, aussi bien au niveau du Job qu’au
|                         | niveau de chaque composant. |

Global Variables

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une
|               | erreur survient. Cette variable est une variable After et
|               | retourne une chaîne de caractères. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres
|                    | composants Azure Storage.
|                    | Des connaissances de Microsoft Azure Storage sont
|                    | nécessaires. |

Scénario associé

Pour des scénarios associés, consultez :

- Scénario : Récupérer des fichiers d’un conteneur Azure Storage à la page 304,
- Scénario : Créer un conteneur dans Azure Storage à la page 286.
- Gérer les données avec Microsoft Azure Table Storage à la page 315
**tAzureStorageContainerCreate**

Ce composant crée un nouveau conteneur de stockage utilisé pour stocker des blobs Azure (Binary Large Object) pour un compte Azure Storage donné.

**Propriétés du tAzureStorageContainerCreate Standard**

Ces propriétés sont utilisées pour configurer le tAzureStorageContainerCreate s’exécutant dans le framework de Jobs Standard.

Le composant tAzureStorageContainerCreate Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Type</strong></td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n'est pas disponible lorsqu'un autre composant de connexion est sélectionné dans la liste <strong>Connection Component</strong>.</td>
</tr>
<tr>
<td><strong>Connection Component</strong></td>
<td>Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage.</td>
</tr>
<tr>
<td><strong>Account Name</strong></td>
<td>Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage.</td>
</tr>
<tr>
<td><strong>Account Key</strong></td>
<td>Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte.</td>
</tr>
<tr>
<td><strong>Protocol</strong></td>
<td>Sélectionnez le protocole de la connexion à créer.</td>
</tr>
</tbody>
</table>
| **Use Azure Shared Access Signature** | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations,
consultez Utilisation des signatures d'accès partagé (SAP).

Dans le champ **Azure Shared Access Signature** qui s'af
diche, saisissez votre signature d'accès partagé entre
guillemets doubles. Vous pouvez obtenir l'URL de la
signature d'accès partagé pour chaque service autorisé
sur le portail de Microsoft Azure, après génération de
la signature. Le format de l'URL est le suivant : https://
<$storagename>..<$service>.core.windows.net/<$sastoken>, où <$storagename> est
le nom du compte de stockage, <$service> le nom
du service autorisé (blob, file, queue ou table) et <$sastoken> est la valeur du jeton de signature
d'accès partagé. Pour plus d'informations, consultez
Constructing the Account SAS URI (en anglais).

Notez que SAS a une période de validité, vous pouvez
configurer l'heure de début de validité de SAS et son
heure d'expiration, après laquelle SAS n'est plus valide
lors de la génération. Vous devez vous assurer que votre
SAS est toujours valide lorsque vous exécutez votre Job.

<table>
<thead>
<tr>
<th>Container name</th>
<th>Saisissez le nom du conteneur de blobs à créer.</th>
</tr>
</thead>
</table>
| Access control | Sélectionnez le niveau de restriction à appliquer sur le
conteneur à créer. |
| Die on error   | Cochez cette case pour arrêter l’exécution du Job
lorsqu’une erreur survient. Décrochez la case pour ignorer les lignes en erreur
et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer
les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de
traitement du Job, aussi bien au niveau du Job qu’au
niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>CONTAINER</th>
<th>Nom du conteneur du blob. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

| Règle d'utilisation | Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job. |
Prérequis
Des connaissances de Microsoft Azure Storage sont nécessaires.

Scénario : Créer un conteneur dans Azure Storage

Dans ce scénario, un Job à quatre composants utilise les composants Azure Storage afin de créer un conteneur dans un système Azure Storage donné et vérifier que le conteneur est bien créé.


Construire le Job

Procédure
1. Dans la perspective Integration du studio, créez un Job vide, nommé azureTalend, par exemple, depuis le nœud Job Designs de la vue Repository.
2. Déposez un tAzureStorageConnection, un tAzureStorageContainerCreate, un tAzureStorageContainerExist et un tJava dans l’espace de modélisation graphique.

Connexion à un compte Azure storage

Procédure
1. Double-cliquez sur le composant tAzureStorageConnection pour ouvrir vue Component.
2. Dans le champ **Account name**, saisissez le nom du compte de stockage auquel se connecter. Dans cet exemple, le compte est *talendstorage*, un compte créé pour la démonstration.

3. Dans le champ **Account key**, collez la clé primaire ou secondaire associée au compte de stockage à utiliser. Ces clés se trouvent dans le dashboard **Manage Access Key** du système Azure Storage auquel se connecter.


### Créer un conteneur

** Procédure **

1. Double-cliquez sur **tAzureStorageContainerCreate** pour ouvrir sa vue **Component**.

![tAzureStorageContainerCreate_1](image)

2. Cochez la case **Use an existing connection** et sélectionnez la connexion précédemment configurée. Dans cet exemple, sélectionnez *tAzureStorageConnection_1*.

3. Dans le champ **Container name**, saisissez le nom du conteneur à créer. Si un conteneur existe avec le même nom, ce conteneur est écrasé lors de l'exécution.

4. Dans la liste **Access control**, sélectionnez le niveau de restriction sur le conteneur à créer. Dans cet exemple, sélectionnez *Private*.

### Vérifier la création

** Procédure **

1. Double-cliquez sur le **tAzureStorageContainerExist** pour ouvrir sa vue **Component**.

![tAzureStorageContainerExist_1](image)

2. Cochez la case **Use an existing connection** et sélectionnez la connexion précédemment configurée. Dans cet exemple, sélectionnez *tAzureStorageConnection_1*.

3. Dans le champ **Container name**, saisissez le nom du conteneur duquel vérifier l'existence.

4. Double-cliquez sur le **tJava** pour ouvrir sa vue **Component**.
5. Dans le champ **Code**, saisissez : `System.out.println();`

6. Dans le panneau **Outline**, par défaut sur la gauche de la vue **Component**, développez le nœud `tAzureStorageContainerExist`.

7. Dans le panneau **Outline**, déposez la variable globale **CONTAINER_EXSIT** dans les parenthèses du code, dans la vue **Component** vue in order to make the code read: `System.out.println(((Boolean)globalMap.get("tAzureStorageContainerExist_1_CONTAINER_EXIST"));

**Exécuter le Job**

**Procédure**

Appuyez sur **F6** pour exécuter ce Job.

**Résultats**

Cela fait, la vue **Run** s’ouvre automatiquement. Vous pouvez examiner les résultats d’exécution.
Vous pouvez constater que le Job retourne true comme résultat de la vérification, ce qui signifie que le conteneur talendcontainer a été créé dans le compte de stockage utilisé.

Dans la console Web du compte Azure storage, vous pouvez également constater que le conteneur talendcontainer a été créé.
tAzureStorageContainerDelete

Ce composant automatise la suppression d’un conteneur de blobs donné dans l’espace d’un compte de stockage spécifique.

Propriétés du tAzureStorageContainerDelete Standard

Ces propriétés sont utilisées pour configurer le tAzureStorageContainerDelete s’exécutant dans le framework de Jobs Standard.

Le composant tAzureStorageContainerDelete Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste <strong>Connection Component</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connection Component</th>
<th>Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage.</th>
</tr>
</thead>
</table>

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations, |
consultez Utilisation des signatures d'accès partagé (SAP).

Dans le champ **Azure Shared Access Signature** qui s'affiche, saisissez votre signature d'accès partagé entre guillemets doubles. Vous pouvez obtenir l'URL de la signature d'accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l'URL est le suivant : https://<$storagename>..<$service>.core.windows.net/<$sastoken>, où <$storagename> est le nom du compte de stockage, <$service> le nom du service autorisé (blob, file, queue ou table) et <$sastoken> est la valeur du jeton de signature d'accès partagé. Pour plus d'informations, consultez Constructing the Account SAS URI (en anglais).

Notez que SAS a une période de validité, vous pouvez configurer l'heure de début de validité de SAS et son heure d'expiration, après laquelle SAS n'est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

<table>
<thead>
<tr>
<th><strong>Container name</strong></th>
<th>Saisissez le nom du conteneur de blobs à supprimer.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Die on error</strong></td>
<td>Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien <strong>Row &gt; Reject</strong>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| **tStatCatcher Statistics** | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. |

**Global Variables**

| **CONTAINER** | Nom du conteneur du blob. Cette variable est une variable **After** et retourne une chaîne de caractères. |
| **ERROR_MESSAGE** | Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. |

**Utilisation**

| **Règle d'utilisation** | Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job. |
| **Prérequis** | Des connaissances de Microsoft Azure Storage sont nécessaires. |
Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tAzureStorageContainerExist

Ce composant automatise la vérification de l’existence d’un conteneur de blobs donné au sein d’un compte de stockage.

Propriétés du tAzureStorageContainerExist Standard

Ces propriétés sont utilisées pour configurer le tAzureStorageContainerExist s’exécutant dans le framework de Jobs Standard.
Le composant tAzureStorageContainerExist Standard appartient à la famille Cloud.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Built-In</td>
<td>Les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>• Repository</td>
<td>Les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connection Component</th>
<th>Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage.</th>
</tr>
</thead>
</table>

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations, |
Dans le champ **Azure Shared Access Signature** qui s’affiche, saisissez votre signature d’accès partagé entre guillemets doubles. Vous pouvez obtenir l’URL de la signature d’accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l’URL est le suivant : https://<storagename>.<service>.core.windows.net/<sastoken>, où <storagename> est le nom du compte de stockage, <service> le nom du service autorisé (blob, file, queue ou table) et <sastoken> est la valeur du jeton de signature d’accès partagé. Pour plus d’informations, consultez [Constructing the Account SAS URI](en anglais).

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

---

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. |

---

**Global Variables**

| CONTAINER | Nom du conteneur du blob. Cette variable est une variable *After* et retourne une chaîne de caractères. |
| CONTAINER_EXIST | Indication de l’existence ou non du conteneur donné. Cette variable est une variable *After* et retourne un booléen. |
| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. |

---

**Utilisation**

| Règle d’utilisation | Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job. |
| Prérequis | Des connaissances de Microsoft Azure Storage sont nécessaires. |

**Scénario associé**

Pour un scénario associé, consultez *Scénario : Créer un conteneur dans Azure Storage* à la page 286.
tAzureStorageContainerList

Ce composant liste tous les conteneurs dans un compte Azure Storage donné.

Propriétés du tAzureStorageContainerList Standard

Ces propriétés sont utilisées pour configurer le tAzureStorageContainerList s’exécutant dans le framework de Jobs Standard.

Le composant tAzureStorageContainerList Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste <strong>Connection Component</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connection Component</th>
<th>Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Account Name</th>
<th>Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Account Key</th>
<th>Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Sélectionnez le protocole de la connexion à créer.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Use Azure Shared Access Signature</th>
<th>Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations,</th>
</tr>
</thead>
</table>
consultez [Utilisation des signatures d'accès partagé (SAP)](https://
<br>&lt;$storagename&gt;.&lt;$service&gt;.core.windows.net/&lt;$sastoken&gt;), où &lt;$storagename&gt; est le nom du compte de stockage, &lt;$service&gt; le nom du service autorisé (blob, file, queue ou table) et &lt;$sastoken&gt; est la valeur du jeton de signature d'accès partagé. Pour plus d'informations, consultez [Constructing the Account SAS URI](https://
<br>en anglais).

Notez que SAS a une période de validité, vous pouvez configurer l'heure de début de validité de SAS et son heure d'expiration, après laquelle SAS n'est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le [Guide utilisateur du Studio Talend](https://
<br>

Le schéma de ce composant est prédéfini avec une seule colonne `ContainerName` de type `String` qui indique le nom de chaque conteneur à lister.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Die on error

Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient.
**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>NB_LINE</th>
<th>Nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prérequis</td>
<td>Des connaissances de Microsoft Azure Storage sont nécessaires.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tAzureStorageDelete

Ce composant supprime des blobs d’un conteneur donné pour un compte Azure Storage, selon les filtres de blobs spécifiés.

Propriétés du tAzureStorageDelete Standard

Ces propriétés sont utilisées pour configurer le tAzureStorageDelete s’exécutant dans le framework de Jobs Standard.
Le composant tAzureStorageDelete Standard appartient à la famille Cloud.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Built-In</td>
<td>les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>• Repository</td>
<td>les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connection Component</th>
<th>Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Name</td>
<td>Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage.</td>
</tr>
<tr>
<td>Account Key</td>
<td>Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte.</td>
</tr>
<tr>
<td>Protocol</td>
<td>Sélectionnez le protocole de la connexion à créer.</td>
</tr>
<tr>
<td>Use Azure Shared Access Signature</td>
<td>Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations,</td>
</tr>
</tbody>
</table>
Dans le champ *Azure Shared Access Signature* qui s’afiche, saisissez votre signature d’accès partagé entre guillemets doubles. Vous pouvez obtenir l’URL de la signature d’accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l’URL est le suivant : `https://<$storagename>.<$service>.core.windows.net/<$sastoken>`, où `<$storagename>` est le nom du compte de stockage, `<$service>` le nom du service autorisé (blob, file, queue ou table) et `<$sastoken>` est la valeur du jeton de signature d’accès partagé. Pour plus d’informations, consultez *Constructing the Account SAS URI* (en anglais).

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

---

<table>
<thead>
<tr>
<th>Container name</th>
<th>Saisissez le nom du conteneur duquel vous souhaitez sélectionner les blobs à lister.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blob filter</td>
<td>Renseignez cette table afin de sélectionner les blobs à lister. Les paramètres à fournir sont :</td>
</tr>
<tr>
<td></td>
<td>• <strong>Blob prefix</strong> : saisissez le préfixe commun au nom des blobs à lister. Ce préfixe vous permet de filtrer les blobs ayant un préfixe spéciﬁé, dans leur nom, dans le conteneur donné.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Include subdirectories</strong> : cochez cette case afin de sélectionner tous les sous-dossiers et les blobs dans ces dossiers, sous le niveau du répertoire désigné. Si vous laissez cette case décochée, le <strong>tAzureStorageList</strong> retourne uniquement les blobs, s’il y en a, directement sous le niveau de ce répertoire.</td>
</tr>
</tbody>
</table>

| Die on error | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**. |
**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>CONTAINER</th>
<th>Nom du conteneur du blob. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prérequis</td>
<td>Des connaissances de Microsoft Azure Storage sont nécessaires.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tAzureStorageGet

Ce composant récupère des blobs d’un conteneur donné pour un compte Azure Storage, en respectant les filtres appliqués sur la hiérarchie virtuelle des blobs, puis écrit les blobs sélectionnés dans un dossier local.

Propriétés du tAzureStorageGet Standard

Ces propriétés sont utilisées pour configurer le tAzureStorageGet s’exécutant dans le framework de Jobs Standard.

Le composant tAzureStorageGet Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.

| Connection Component | Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage. |

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans |
utiliser de clé de compte. Pour plus d’informations, consultez **Utilisation des signatures d’accès partagé (SAP).**

Dans le champ **Azure Shared Access Signature** qui s’affiche, saisissez votre signature d’accès partagé entre guillemets doubles. Vous pouvez obtenir l’URL de la signature d’accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l’URL est le suivant : `https://<$storagename>.$<service>.core.windows.net/<$sastoken>`, où `<$storagename>` est le nom du compte de stockage, `<$service>` le nom du service autorisé (blob, file, queue ou table) et `<$sastoken>` est la valeur du jeton de signature d’accès partagé. Pour plus d’informations, consultez **Constructing the Account SAS URI (en anglais).**

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

<table>
<thead>
<tr>
<th>Container</th>
<th>Saisissez le nom du conteneur duquel récupérer les blobs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local folder</td>
<td>Saisissez le chemin d’accès ou parcourez votre système jusqu’au dossier dans lequel vous souhaitez stocker les blobs récupérés.</td>
</tr>
</tbody>
</table>
| Blobs | Renseignez cette table afin de sélectionner les blobs à récupérer. Les paramètres à fournir sont :

  - **Prefix** : saisissez le préfixe commun au nom des blobs à lister. Ce préfixe vous permet de filtrer les blobs ayant un préfixe spécifié, dans leur nom, dans le conteneur donné.

    Le nom d’un blob contient la hiérarchie virtuelle du blob lui-même. Cette hiérarchie est un chemin virtuel vers ce blob et est relatif au conteneur où est stocké le blob. Par exemple, dans un conteneur nommé `photos`, le nom d’un blob d’une photo peut être `2014/US/Oakland/Talend.jpg`.


    Si vous souhaitez sélectionner les blobs stockés directement sous le niveau du conteneur, c’est-à-dire les blobs sans chemin virtuel dans leur nom, supprimez les guillemets et saisissez `null`.

  - **Include-subdirectories** : cochez cette case afin de sélectionner tous les sous-dossiers et les blobs dans ces dossiers, sous le niveau du répertoire désigné. Si vous laissez cette case décochée, le `tAzureStorageList` retourne uniquement les blobs, s’il y en a, directement sous le niveau de ce répertoire.
**Create parent directories**: cochez cette case pour répliquer le répertoire virtuel des blobs récupérés dans le dossier local.

Notez que si vous laissez cette case décochée, le répertoire doit être le même dans le dossier local que celui des blobs récupérés dans le conteneur. Sinon, ces blobs ne pourront être récupérés.

**Die on error**

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Global Variables**

| CONTAINER | Nom du conteneur du blob. Cette variable est une variable **After** et retourne une chaîne de caractères. |
| LOCAL_FOLDER | Répertoire local utilisé dans ce composant. Cette variable est une variable **After** et retourne une chaîne de caractères. |
| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. |

**Utilisation**

**Règle d’utilisation**

Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job.

**Prérequis**

Des connaissances de Microsoft Azure Storage sont nécessaires.

**Scénario : Récupérer des fichiers d'un conteneur Azure Storage**

Dans ce scénario, un Job a cinq composants utilise des composants Azure Storage pour écrire des fichiers dans un système Azure Storage et récupérer les fichiers sélectionnés (des blobs, dans le vocabulaire Azure Storage) de ce système.

Le conteneur talendcontainer utilisé dans ce scénario a été créé à l’aide du composant tAzureStorageContainerCreate, dans le scénario Scénario : Créer un conteneur dans Azure Storage à la page 286.

**Construire le Job**

**Procédure**

1. Dans la perspective Integration du Studio, créez un Job vide, nommé azureTalend par exemple, depuis le nœud Job Designs de la vue Repository.
2. Déposez un tAzureStoragePut, un tAzureStorageList, un tJava et un tAzureStorageGet dans l’espace de modélisation graphique.

**Connexion à un compte Azure Storage**

**Procédure**

1. Double-cliquez sur le tAzureStorageConnection pour ouvrir sa vue Component.
2. Dans le champ Account name, saisissez le nom du compte de stockage auquel se connecter. Dans cet exemple, saisissez talendstorage, un compte créé pour la démonstration.

3. Dans le champ Account key, collez la clé primaire ou secondaire associée au compte de stockage à utiliser. Ces clés se trouvent dans le dashboard Manage Access Key du système Azure Storage auquel se connecter.

4. Dans la liste Protocol, sélectionnez le protocole pour l’endpoint du compte de stockage à utiliser. Dans cet exemple, sélectionnez HTTPS.

**Ecrire des fichiers dans Azure Storage**

**Procédure**

1. Double-cliquez sur le tAzureStoragePut afin d’ouvrir sa vue Component.

2. Cochez la case Use an existing connection puis sélectionnez la connexion précédemment configurée. Dans cet exemple, sélectionnez tAzureStorageConnection_1.

3. Dans le champ Container name, saisissez le nom du conteneur dans lequel écrire les fichiers. Dans cet exemple, le conteneur est talendcontainer, un conteneur créé dans le scénario Scénario : Créer un conteneur dans Azure Storage à la page 286.

4. Dans le champ Local folder, saisissez le chemin d’accès ou parcourez votre système jusqu’au répertoire où sont stockés les fichiers à utiliser. Dans ce scénario, des images de processus techniques sont stockées localement dans E:/photos. Saisissez E:/photos ; cela permet au tAzureStoragePut de charger tous les fichiers du dossier et des sous-dossiers dans le conteneur talendcontainer.

Dans un but de démonstration, les photos d’exemple sont organisées comme suit, dans le dossier E:/photos :

- Directement sous le niveau E:/photos :
Dans le champ **Azure Storage folder**, saisissez le chemin d’accès au répertoire dans lequel vous souhaitez écrire les fichiers. Ce répertoire est directement créé dans le conteneur à utiliser, s’il n’existe pas. Dans cet exemple, saisissez *photos*.

Si vous ne saisissez rien et que vous laissez les guillemets par défaut, les fichiers et leur répertoire local seront écrits directement sous le niveau du conteneur.

**Vérifier le transfert des fichiers**

**Configurer le tAzureStorageList**

**Procédure**

1. Double-cliquez sur le **tAzureStorageList** pour ouvrir sa vue **Component**.

2. Cochez la case **Use an existing connection** et sélectionnez la connexion précédemment configurée. Dans cet exemple, sélectionnez **tAzureStorageConnection_1**.
3. Dans le champ **Container name**, saisissez le nom du conteneur dans lequel vous devez vérifier que les fichiers existent. Dans ce scénario, saisissez **talendcontainer**.

4. Sous la table **Blob filter**, cliquez sur le bouton **[+]** pour ajouter une ligne à la table.

5. Dans la colonne **Prefix**, saisissez le préfixe commun au nom des fichiers (blobs) à vérifier. Ce préfixe représente un répertoire virtuel que vous désignez comme point de départ de vérification des fichiers (blobs). Dans cet exemple, saisissez **photos/**.


6. Dans la colonne **Include sub-directories**, cochez la case dans la ligne ajoutée. Cela permet au **tAzureStorageList** de vérifier tous les fichiers quel que soit leur niveau hiérarchique sous le point de départ désigné.

###Configurer le tJava

**Procédure**

1. Double-cliquez sur le **tJava** afin d’ouvrir sa vue **Component**.

![Image](image1.png)

2. Dans le champ **Code**, saisissez : `System.out.println();`

3. Dans le panneau **Outline**, par défaut à gauche de la vue **Component**, développez le nœud **tAzureStorageList**.

![Image](image2.png)

4. Dans le panneau **Outline**, déposez la variable globale **CONTAINER_BLOB** entre les parenthèses du code dans la vue **Component** afin que le code soit le suivant : `System.out.println(((Boolean)globalMap.get("tAzureStorageList_1_CURRENT_BLOB")));`

###Récupérer les fichiers sélectionnés

**Procédure**

1. Double-cliquez sur le **tAzureStorageGet** pour ouvrir sa vue **Component**.
2. Cochez la case **Use an existing connection** et sélectionnez la connexion précédemment configurée. Dans cet exemple, sélectionnez **tAzureStorageConnection_1**.

3. Dans le champ **Container name**, saisissez le nom du conteneur duquel récupérer les fichiers. Dans ce scénario, saisissez **talendcontainer**.

4. Dans le champ **Local folder**, saisissez le chemin d’accès ou parcourrez votre système jusqu’au répertoire où stocker les fichiers récupérés. Dans cet exemple, saisissez **E:/screenshots**.

5. Sous la table **Blob**, cliquez le bouton [+] pour ajouter une ligne à la table.

6. Dans la colonne **Prefix**, saisissez le préfixe du nom commun aux fichiers (blobs) à récupérer. Dans cet exemple, saisissez **photos/mongodb/**.

7. Dans la colonne **Include sub-directories**, cochez la case dans la nouvelle ligne. Cela permet au **tAzureStorageGet** de récupérer tous les fichiers (blobs) sous le niveau **photos/mongodb/**.

8. Dans la colonne **Create parent directories**, cochez la case dans la ligne ajoutée afin de créer le même répertoire dans le dossier local spécifié que pour les blobs récupérés dans le conteneur. Notez que le fait d’avoir le même répertoire est nécessaire pour récupérer les blobs. Si vous laissez cette case décochée, vous devez créer le même répertoire vous-même dans le dossier local cible.

**Exécuter le Job**

**Procédure**

Appuyez sur **F6** pour exécuter ce Job.

**Résultats**

Cela fait, la vue **Run** s’ouvre automatiquement. Vous pouvez examiner les résultats d’exécution.
Vous pouvez constater que le Job retourne la liste des blobs, ainsi que le préfixe *photos* dans le conteneur.

Vous pouvez également le constater dans la console Web du compte de stockage Azure :

Dans le dossier local spécifié, les blobs ayant le préfixe *photos/mongodb/* ont été récupérés. Leur préfixe a été transformé en répertoire.
tAzureStorageInputTable

Ce composant récupère, depuis une table Azure Storage, un ensemble d’entités répondant aux critères de filtre spécifiés.

Propriétés du tAzureStorageInputTable Standard

Ces propriétés sont utilisées pour configurer le tAzureStorageInputTable s’exécutant dans le framework de Jobs Standard.

Le composant tAzureStorageInputTable Standard appartient à la famille Cloud.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage. |

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans |
utiliser de clé de compte. Pour plus d'informations, consultez **Utilisation des signatures d'accès partagé (SAP)**.

Dans le champ **Azure Shared Access Signature** qui s'affiche, saisissez votre signature d'accès partagé entre guillemets doubles. Vous pouvez obtenir l’URL de la signature d'accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l’URL est le suivant : `https://<$storagename>.<$service>.core.windows.net/<$sastoken>`.

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

<table>
<thead>
<tr>
<th>Table name</th>
<th>Spécifiez le nom de la table de laquelle récupérer les entités.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schema et Edit schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <em>line</em> lors du nominal des champs.</td>
</tr>
</tbody>
</table>

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le [Guide utilisateur du Studio Talend](#).

- **Repository** : Le schéma existe déjà et il est stocké dans le [Repository](#). Ainsi, il peut être réutilisé. Voir également le [Guide utilisateur du Studio Talend](#).

Le schéma de ce composant est prédéfini et contient les colonnes suivantes décrivant les trois propriétés système de chaque entité :

- **PartitionKey** : clé de partition de la partition à laquelle l’entité appartient.

- **RowKey** : clé de ligne pour l’entité au sein de la partition.

  **PartitionKey** et **RowKey** sont des valeurs de type String identifiant de manière unique chaque entité dans une table. L’utilisateur doit les inclure dans chaque opération insert, update, ou delete (insertion, mise à jour ou suppression).

- **Timestamp** : heure de dernière modification de l’entité. Cette valeur DateTime est maintenue par le serveur Azure et ne peut être modifiée par l’utilisateur.
Pour plus d’informations concernant ces propriétés système, consultez [Understanding the Table Service Data Model](en anglais).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Use filter expression

Cochez cette case et renseignez la table **Filter expressions** affichée, afin de spécifier les conditions utilisées pour filtrer les entités à récupérer, en cliquant autant de fois que nécessaire sur le bouton `[+]`, afin d’ajouter une ligne par condition. Configurez la valeur des paramètres suivants pour chaque condition.

- **Column** : spécifiez le nom de la propriété sur laquelle appliquer la condition.
- **Function** : cliquez dans la cellule et sélectionnez dans la liste l’opérateur de comparaison à utiliser.
- **Value** : spécifiez la valeur utilisée pour la comparaison de la propriété.
- **Predicate** : sélectionnez le prédicat utilisé pour combiner les conditions.
- **Field type** : cliquez dans la cellule et sélectionnez dans la liste déroulante le type de la colonne.

L’expression de filtre générée s’affiche dans le champ **Effective filter**, en lecture seule.

Pour plus d’informations concernant les expressions de filtres, consultez [Querying Tables and Entities](en anglais).

### Die on error

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

### Advanced settings

### Name mappings

Renseignez cette table pour mapper le nom de la colonne du schéma du composant au nom de la propriété de l’entité de la table Azure s’ils sont différents.
**Gérer les données avec Microsoft Azure Table Storage**

Voici un exemple d’utilisation des composants Talend pour vous connecter à un compte Microsoft Azure Storage, vous donnant accès à un service de tables de stockage Azure, écrire des données relatives à des employés dans une table Azure et récupérer les données des employés depuis la table, pour les afficher dans la console.

Les données relatives aux employés utilisées dans cet exemple se présentent comme suit :

```
#Id; Name; Site; Job; Date; Salary
12000; Gerald Roosevelt; Beijing; Software Developer; 2008-01-01; 15000.01
12001; Benjamin Harrison; Paris; Software Developer; 2008-11-22; 13000.11
12002; Bob Clinton; Beijing; Software Tester; 2008-05-12; 12000.22
12003; James Quincy; Paris; Technical Writer; 2009-03-10; 12000.33
12004; Gerald Harrison; Beijing; Software Tester; 2009-06-20; 12500.44
```
Création d’un Job pour gérer des données avec Azure Table Storage

Créez un Job pour vous connecter à un compte Azure Storage, écrire des données relatives à des employés, dans une table Azure Storage puis récupérer ces informations de la table et les afficher dans la console.

Procédure

1. Créez un nouveau Job et ajoutez un composant `tAzureStorageConnection`, un `tFixedFlowInput`, un `tAzureStorageOutputTable`, un `tAzureStorageInputTable` et un `tLogRow`, en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le `tFixedFlowInput` au `tAzureStorageOutputTable` à l’aide d’un lien `Row > Main`.
3. Répétez cette étape pour relier le `tAzureStorageInputTable` au `tLogRow`.
4. Reliez le composant `tAzureStorageConnection` au `tFixedFlowInput`, à l’aide d’un lien `Trigger > OnSubjobOk`.
5. Répétez l’opération pour relier le `tFixedFlowInput` au `tAzureStorageInputTable`.

Connexion à un compte Azure Storage

Configurez le composant `tAzureStorageConnection` pour ouvrir la connexion à un compte Azure Storage.
Avant de commencer

Le compte Azure Storage, qui vous permet d’accéder au service Azure Table Storage et de stocker les données fournies relatives aux employés doit déjà avoir été créé. Pour plus d’informations concernant la création d’un compte Azure Storage, consultez À propos des comptes de stockage Azure.

Procédure

1. Double-cliquez sur le composant **tAzureStorageConnection** pour ouvrir sa vue **Basic settings** dans l’onglet **Component**.

2. Dans le champ **Account Name**, spécifiez le nom du compte auquel vous souhaitez accéder.

3. Dans le champ **Account Key**, spécifiez la clé associée au compte auquel accéder.

Écriture de données dans une table Azure Storage

Configurez le **tFixedFlowInput** et le **tAzureStorageOutputTable** pour écrire les données relatives aux employés dans une table Azure Storage.

Procédure

1. Double-cliquez sur le composant **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur **Edit schema** à côté du champ et définissez le schéma dans la fenêtre qui s’ouvre, en ajoutant six colonnes : **Id**, **Name**, **Site**, et **Job** de type **String**, **Date** de type **Date** et **Salary**, de type **Double**. Cliquez sur **OK** afin de sauvegarder les modifications et acceptez la propagation proposée par la boîte de dialogue.
Notez que, dans cet exemple, les colonnes Site et Id sont utilisées pour alimenter les valeurs des propriétés système PartitionKey et RowKey de chaque entité et doivent être de type String. La colonne Name est utilisée pour alimenter la valeur de la propriété EmployeeName de chaque entité.

3. Dans la zone Mode, sélectionnez Use Inline Content delimited file et, dans le champ Content qui s’affiche, saisissez les données relatives aux employés qui seront écrites dans la table Azure Storage.

4. Double-cliquez sur le composant tAzureStorageOutputTable pour ouvrir sa vue Basic settings.

5. Dans la liste déroulante des composants de connexion, sélectionnez le composant dont les informations de connexion seront utilisées pour configurer la connexion au service Azure Storage, tAzureStorageConnection_1 dans cet exemple.

6. Dans le champ Table name, saisissez le nom de la table dans laquelle les données relatives aux employés seront écrites, employee dans cet exemple.

7. Dans la liste Action on table, sélectionnez l’opération à effectuer sur une table spécifiée, Drop table if exist and create, dans cet exemple.

9. Cliquez trois fois sur le bouton sous la table **Name mappings**, pour ajouter trois lignes et mapper le nom de la colonne du schéma au nom de propriété de chaque entité dans la table Azure table. Dans cet exemple,

- la colonne Site est utilisée pour alimenter la valeur de la propriété système **PartitionKey**. Dans la première ligne, vous devez configurer la cellule **Schema column name** avec la valeur "Site" et la cellule **Entity property name** avec la valeur "PartitionKey".
- la colonne Id est utilisée pour alimenter la valeur de la propriété système **RowKey**. Dans la deuxième ligne, vous devez configurer la cellule **Schema column name** avec la valeur "Id" et la cellule **Entity property name** avec la valeur "RowKey".
- la colonne Name est utilisée pour alimenter la valeur de la propriété **EmployeeName**. Dans la troisième ligne, vous devez configurer la cellule **Schema column name** avec la valeur "Name" et la cellule **Entity property name** avec la valeur "EmployeeName".

**Récupération de données de la table Azure Storage**

Configurez le composant **tAzureStorageInputTable** et le **tLogRow** pour récupérer les données relatives aux employés, depuis la table Azure Storage.

**Procédure**

1. Double-cliquez sur le composant **tAzureStorageInputTable** pour ouvrir sa vue **Basic settings**.

2. Dans la liste déroulante des composants de connexion, sélectionnez le composant duquel utiliser les informations de connexion pour configurer la connexion au service Azure Storage, **tAzureStorageConnection_1** dans cet exemple.

3. Dans le champ **Table name**, saisissez le nom de la table de laquelle récupérer les données relatives aux employés, **employee** dans cet exemple.

4. Cliquez sur le bouton à côté du champ **Edit schema** pour ouvrir la fenêtre de définition du schéma.
Notez que ce schéma a déjà été prédéfini et contient deux colonnes en lecture seule RowKey et PartitionKey de type String et une autre colonne Timestamp de type Date. Les colonnes RowKey et PartitionKey correspondent aux colonnes Id et Site du schéma du tAzureStorageO utputTable.

5. Définissez le schéma en ajoutant quatre colonnes contenant les autres informations relatives aux employés, Name et Job de type String, Date de type Date et Salary de type Double. Cliquez sur OK afin de sauvegarder les modifications et acceptez la propagation proposée par la boîte de dialogue.


7. Cliquez sur le bouton sous la table Name mappings, pour ajouter une ligne. Dans Schema column name, saisissez "Name" et, dans Entity property name, saisissez "EmployeeName" pour mapper le nom de la colonne au nom de la propriété de chaque entité dans la table Azure.

Notez que pour le composant tAzureStorageInputTable, les colonnes PartitionKey et RowKey ont déjà été ajoutées automatiquement au schéma et vous n'avez pas besoin de spécifier la relation de mapping pour ces colonnes.

8. Double-cliquez sur le composant tLogRow pour ouvrir sa vue Basic settings et, dans la zone Mode, sélectionnez Table (print values in cells of a table) pour un affichage optimal des résultats.
Exécution du Job afin de gérer les données avec Azure Table Storage

Après avoir configuré le Job et ses composants pour gérer les données avec Azure Table Storage, vous pouvez exécuter le Job et vérifier les résultats d’exécution du Job.

Procédure

1. Appuyez sur les touches Ctrl + S afin de sauvegarder le Job.

Comme affiché ci-dessus, le Job est bien exécuté et les données relatives aux employés sont affichées dans la console, avec l’horodatage indiquant le moment d’insertion de chaque entité.

3. Vérifiez que les données relatives aux employés ont bien été écrites dans la table Azure Storage employee, en utilisant Microsoft Azure Storage Explorer, si vous le souhaitez.
**tAzureStorageList**

Ce composant liste des blobs dans un conteneur donné, selon les filtres de blobs spécifiés.

**Propriétés du tAzureStorageList Standard**

Ces propriétés sont utilisées pour configurer le tAzureStorageList s’exécutant dans le framework de Jobs Standard.

Le composant tAzureStorageList Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Type</strong></td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste <strong>Connection Component</strong>.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage. |

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations, |
Dans le champ **Azure Shared Access Signature** qui s’affiche, saisissez votre signature d’accès partagé entre guillemets doubles. Vous pouvez obtenir l’URL de la signature d’accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l’URL est le suivant : `https://<storagename>.<service>.core.windows.net/<sastoken>`, où `<storagename>` est le nom du compte de stockage, `<service>` le nom du service autorisé (blob, file, queue ou table) et `<sastoken>` est la valeur du jeton de signature d’accès partagé. Pour plus d’informations, consultez *Constructing the Account SAS URI* (en anglais).

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

<table>
<thead>
<tr>
<th>Container name</th>
<th>Saisissez le nom du conteneur duquel vous souhaitez sélectionner les blobs à lister.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blob filters</td>
<td>Renseignez cette table afin de sélectionner les blobs à lister. Les paramètres à fournir sont :</td>
</tr>
<tr>
<td></td>
<td>• <strong>Prefix</strong> : saisissez le préfixe commun au nom des blobs à lister. Ce préfixe vous permet de filtrer les blobs ayant un préfixe spécifié, dans leur nom, dans le conteneur donné.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Include-subdirectories</strong> : cochez cette case afin de sélectionner tous les sous-dossiers et les blobs dans ces dossiers, sous le niveau du répertoire désigné. Si vous laissez cette case décochée, le tAzureStorageList retourne uniquement les blobs, s’il y en a, directement sous le niveau de ce répertoire.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs.</td>
</tr>
</tbody>
</table>
• **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

• **Repository** : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Le schéma de ce composant est prédéfini avec une colonne unique `BlobName` de type `String`, qui indique le nom de chaque blob à lister.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

• **View schema** : sélectionnez cette option afin de voir le schéma.

• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métdonnée du schéma dans la fenêtre [Repository Content].

### Die on error

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject.

### Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les métdonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>CONTAINER</th>
<th>Nom du conteneur du blob. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURRENT_BLOB</td>
<td>Nom du blob traité par ce composant. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
</tbody>
</table>
ERROR_MESSAGE
Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prérequis</td>
<td>Des connaissances de Microsoft Azure Storage sont nécessaires.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez **Scénario : Récupérer des fichiers d’un conteneur Azure Storage** à la page 304.
tAzureStorageOutputTable

Ce composant effectue l’action définie sur une table Azure Storage et insère, remplace, fusionne ou supprime des entités dans la table, en se basant sur les données entrantes du composant précédent.

Propriétés du tAzureStorageOutputTable Standard

Ces propriétés sont utilisées pour configurer le tAzureStorageOutputTable s’exécutant dans le framework de Jobs Standard.

Le composant tAzureStorageOutputTable Standard appartient à la famille Cloud.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>Repository</td>
<td>les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage. |

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans |
utiliser de clé de compte. Pour plus d’informations, consultez Utilisation des signatures d’accès partagé (SAP).

Dans le champ Azure Shared Access Signature qui s’affiche, saisissez votre signature d’accès partagé entre guillemets doubles. Vous pouvez obtenir l’URL de la signature d’accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l’URL est le suivant : https://<$storagename>.<$service>.core.windows.net/<$sastoken>, où <$storagename> est le nom du compte de stockage, <$service> le nom du service autorisé (blob, file, queue ou table) et <$sastoken> est la valeur du jeton de signature d’accès partagé. Pour plus d’informations, consultez Constructing the Account SAS URI (en anglais).

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

**Table name**
Spécifiez le nom de la table dans laquelle les entités seront écrites.

**Schema et Edit schema**
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Partition Key**
 Sélectionnez dans la liste déroulante la colonne du schéma contenant la valeur de la clé de partition.
<table>
<thead>
<tr>
<th>Row Key</th>
<th>Sélectionnez dans la liste la colonne du schéma contenant la valeur de la clé de ligne.</th>
</tr>
</thead>
</table>
| Action on data | Sélectionnez une action à effectuer sur les données de la table définie.  
• **Insert** : insère une nouvelle entité à la table.  
• **Insert or replace** : remplace une entité existante ou insère une nouvelle entité si elle n’existe pas. Lorsqu’une entité est remplacée, les propriétés de l’entité précédente sont supprimées si la nouvelle entité ne les définit pas.  
• **Insert or merge** : fusionne une entité existante ou insère une nouvelle entité si elle n’existe pas. Lors de la fusion d’une entité, les propriétés de l’entité précédente seront conservées si la nouvelle entité ne les définit pas ou ne les inclut pas.  
• **Merge** : met à jour une entité existante sans supprimer la valeur de l’entité précédente si la nouvelle entité ne définit pas sa valeur.  
• **Replace** : met à jour une entité existante et supprime la valeur de la propriété de l’entité précédente si la nouvelle entité ne définit pas sa valeur.  
• **Delete** : supprime une entité existante.  
Pour des raisons de performance, les données entrantes sont traitées en parallèle et en ordre aléatoire. Il n’est donc pas recommandé d’effectuer des opérations sensibles à l’ordre (par exemple une insertion ou un replacement), s’il y a des lignes en doublon dans vos données. |
| Action on table | Sélectionnez une opération à effectuer sur la table définie, vous pouvez effectuer l’une des opérations suivantes :  
• **Default** : aucune opération n’est effectuée.  
• **Drop and create table** : la table est supprimée et créée à nouveau.  
• **Create table** : la table n’existe pas et est créée.  
• **Create table if does not exist** : la table est créée si elle n’existe pas.  
• **Drop table if exist and create** : la table est supprimée si elle existe déjà et créée à nouveau. |
| Process in batch | Cochez cette case pour traiter par lots les entités d’entrée.  
Notez que les entités à traiter par lots doivent appartenir au même groupe de partition, ce qui signifie qu’elles doivent avoir la même valeur de clé de partition. |
| Die on error | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. |
**Advanced settings**

### Name mappings

Renseignez cette table pour mapper le nom de la colonne du schéma du composant au nom de la propriété de l’entité de la table Azure s’ils sont différents.

- **Schema column name** : saisissez le nom de la colonne du schéma du composant, entre guillemets doubles.
- **Entity property name** : saisissez le nom de la propriété de l’entité de la table Azure, entre guillemets doubles.

Par exemple, si trois colonnes de schéma `CompanyID`, `EmployeeID` et `EmployeeName` sont utilisées pour alimenter la valeur des propriétés des entités `PartitionKey`, `RowKey` et `Name`, respectivement, vous devez ajouter les lignes suivantes pour le mapping lors de l’écriture des données dans la table Azure.

- dans la cellule **Schema column name**, renseignez la valeur "CompanyID" et, dans la cellule **Entity property name**, renseignez la valeur "PartitionKey".
- dans la cellule **Schema column name**, renseignez la valeur "EmployeeID" et, dans la cellule **Entity property name**, renseignez la valeur "RowKey".
- dans la cellule **Schema column name**, renseignez la valeur "EmployeeName" et, dans la cellule **Entity property name**, renseignez la valeur "Name".

### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_SUCCESS</strong></td>
<td>Nombre de lignes correctement traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_REJECT</strong></td>
<td>Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé comme composant de fin dans un Job ou un sous-job et nécessite un lien d’entrée.
Scénario associé

Pour un scénario associé, consultez Gérer les données avec Microsoft Azure Table Storage à la page 315.
tAzureStoragePut

Ce composant charge des fichiers locaux dans un conteneur donné, pour un compte Azure Storage.

**Propriétés du tAzureStoragePut Standard**

Ces propriétés sont utilisées pour configurer le tAzureStoragePut s’exécutant dans le framework de Jobs Standard.

Le composant tAzureStoragePut Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Type</strong></td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td>• <strong>Built-In</strong></td>
<td>les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>• <strong>Repository</strong></td>
<td>les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage. |

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations, |
consultez [Utilisation des signatures d'accès partagé (SAP)](https://docs.microsoft.com/zh-CN/azure/storage/common/access-keys).

Dans le champ **Azure Shared Access Signature** qui s'affiche, saisissez votre signature d'accès partagé entre guillemets doubles. Vous pouvez obtenir l'URL de la signature d'accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l'URL est le suivant: `https://<$storagename>.<$service>.core.windows.net/<$sastoken>`, où `<$storagename>` est le nom du compte de stockage, `<$service>` le nom du service autorisé (blob, file, queue ou table) et `<$sastoken>` est la valeur du jeton de signature d'accès partagé. Pour plus d'informations, consultez [Constructing the Account SAS URI](https://docs.microsoft.com/zh-CN/azure/storage/common/access-keys).

Notez que SAS a une période de validité, vous pouvez configurer l'heure de début de validité de SAS et son heure d'expiration, après laquelle SAS n'est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

| Container name | Saisissez le nom du conteneur dans lequel écrire les fichiers. Ce conteneur doit exister dans le système Azure Storage que vous utilisez. |
| Local folder | Saisissez le chemin d'accès, ou parcourez votre système jusqu'au dossier duquel vous souhaitez charger les fichiers. |
| Azure storage folder | Saisissez le chemin d'accès au dossier du blob virtuel dans le système Azure distant de stockage dans lequel vous souhaitez charger des fichiers. Si vous ne saisissez aucune valeur dans ce champ et que vous laissez les seuls guillemets par défaut, le composant tAzureStoragePut écrit les fichiers directement au niveau sous le conteneur. |
| Use file list | Cochez cette case afin de définir les conditions de filtre de fichiers. Une fois la case cochée, la table Files s'affiche. |
| Files | Renseignez cette table pour sélectionner les fichiers à charger dans Azure. Les paramètres à fournir sont :
  - **Filemask** : nom ou chemin des fichiers à charger.
  - **New name** : nom à donner aux fichiers après leur chargement. |
| Die on error | Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**. |
Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Global Variables

| CONTAINER | Nom du conteneur du blob. Cette variable est une variable After et retourne une chaîne de caractères. |
| LOCAL_FOLDER | Répertoire local utilisé dans ce composant. Cette variable est une variable After et retourne une chaîne de caractères. |
| REMOTE_FOLDER | Répertoire distant utilisé dans ce composant. Cette variable est une variable After et retourne une chaîne de caractères. |
| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job. |
| Prérequis | Des connaissances de Microsoft Azure Storage sont nécessaires. |

Scénario associé

Pour un scénario associé, consultez Scénario : Récupérer des fichiers d’un conteneur Azure Storage à la page 304.
**tAzureStorageQueueCreate**

Ce composant crée une nouvelle file dans un compte Azure Storage donné.

**Propriétés du tAzureStorageQueueCreate Standard**

Ces propriétés sont utilisées pour configurer le tAzureStorageQueueCreate s’exécutant dans le framework de Jobs Standard.

Le tAzureStorageQueueCreate Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste <strong>Connection Component</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connection Component</th>
<th>Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage.</th>
</tr>
</thead>
</table>

| Account Name                    | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key                     | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol                        | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations, |
consultez Utilisation des signatures d’accès partagé (SAP).

Dans le champ Azure Shared Access Signature qui s’affiche, saisissez votre signature d’accès partagé entre guillemets doubles. Vous pouvez obtenir l’URL de la signature d’accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l’URL est le suivant : https://<$storagename>.<$service>.core.windows.net/<$sastoken>, où <$storagename> est le nom du compte de stockage, <$service> le nom du service autorisé (blob, file, queue ou table) et <$sastoken> est la valeur du jeton de signature d’accès partagé. Pour plus d’informations, consultez Constructing the Account SAS URI (en anglais).

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

**Queue name**
Spécifiez le nom de la file Azure à créer. Pour plus d’informations concernant les conventions de nommage des files, consultez Naming Queues and Metadata (en anglais).

**Die on error**
Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

**Advanced settings**

**tStatCatcher Statistics**
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Global Variables**

**QUEUE_NAME**
Nom de la file Azure. Cette variable est une variable After et retourne une chaîne de caractères.

**ERROR_MESSAGE**
Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

**Utilisation**

Règle d’utilisation
Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job.

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
**tAzureStorageQueueDelete**

Ce composant supprime de manière définitive une file spécifiée pour un compte Azure Storage donné.

**Propriétés du tAzureStorageQueueDelete Standard**

Ces propriétés sont utilisées pour configurer le tAzureStorageQueueDelete s’exécutant dans le framework de Jobs Standard.

Le composant tAzureStorageQueueDelete Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Property Type | Sélectionnez la manière de configurer les informations de connexion.  
- **Built-In** : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.  
- **Repository** : les informations de connexion stockées centralement dans le Repository > Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component. |

| Connection Component | Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage. |

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations, |
consultez Utilisation des signatures d'accès partagé (SAP).

Dans le champ **Azure Shared Access Signature** qui s'af fiche, saisissez votre signature d'accès partagé entre guillemets doubles. Vous pouvez obtenir l'URL de la signature d'accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l'URL est le suivant: https://<$storagename>.<$service>.core.windows.net/<$sastoken>, où <$storagename> est le nom du compte de stockage, <$service> le nom du service autorisé (blob, file, queue ou table) et <$sastoken> est la valeur du jeton de signature d'accès partagé. Pour plus d’informations, consultez **Constructing the Account SAS URI** (en anglais).

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

<table>
<thead>
<tr>
<th>Queue name</th>
<th>Spécifiez le nom de la file Azure à supprimer.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die on error</td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>QUEUE_NAME</th>
<th>Nom de la file Azure. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tAzureStorageQueueInput

Ce composant récupère un ou plusieurs messages du début d’une file Azure.

Propriétés du tAzureStorageQueueInput Standard

Ces propriétés sont utilisées pour configurer le tAzureStorageQueueInput s’exécutant dans le framework de Jobs Standard.

Le composant tAzureStorageQueueInput Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton […] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage. |

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations, |
consultez Utilisation des signatures d'accès partagé (SAP).

Dans le champ Azure Shared Access Signature qui s’affiche, saisissez votre signature d’accès partagé entre guillemets doubles. Vous pouvez obtenir l’URL de la signature d’accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l’URL est le suivant : https://<$storagename>.<$service>.core.windows.net/<$sastoken>, où <$storagename> est le nom du compte de stockage, <$service> le nom du service autorisé (blob, file, queue ou table) et <$sastoken> est la valeur du jeton de signature d’accès partagé. Pour plus d’informations, consultez Constructing the Account SAS URI (en anglais).

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

Queue name
Spécifiez le nom de la file Azure de laquelle récupérer les messages.

Schema et Edit schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

• Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.
• Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Le schéma de ce composant est prédéfini avec les colonnes suivantes :
• MessageId : ID du message.
• MessageContent : corps du message.
• InsertionTime : l’heure à laquelle le message a été ajouté à la file.
• ExpirationTime : l’heure à laquelle le message va expirer.
• NextVisibleTime : l’heure à laquelle le message sera à nouveau visible.
• DequeueCount : nombre de fois que le message a été retiré de la file. Cette valeur est incrémentée chaque fois que le message est retiré de la file, mais n’est pas incrémentée lorsque le message est regardé.
• PopReceipt : valeur de l’accusé réception pop requis pour supprimer le message.
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Number of messages
Saisissez le nombre de messages à récupérer de la file en une fois, jusqu’à 32.

### Peek messages
Cochez cette case pour récupérer les messages sans les supprimer de la file ou altérer leur visibilité. Les messages seront toujours disponibles pour les autres consommateurs.

### Delete the message while streaming
Cochez cette case afin de supprimer les messages en les récupérant de la file.

### Die on error
Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

#### Advanced settings

**Visibility timeout in seconds**
Saisissez (en secondes) la période avant suspension de la visibilité relative à l’heure du serveur. Cette période avant suspension est ajoutée à l’heure à laquelle le message est récupéré, afin de déterminer sa valeur **NextVisibleTime**. Le message ne sera pas visible pour les autres consommateurs durant cet intervalle après récupération.

### tStatCatcher Statistics
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

#### Global Variables

**NB_LINE**
Nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.

**QUEUE_NAME**
Nom de la file Azure. Cette variable est une variable **After** et retourne une chaîne de caractères.
| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie. |

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tAzureStorageQueueInputLoop

Ce composant exécute une boucle sans fin pour récupérer des messages du début d’une file Azure.

Propriétés du tAzureStorageQueueInputLoop Standard

Ces propriétés sont utilisées pour configurer le tAzureStorageQueueInputLoop s’exécutant dans le framework de Jobs Standard Job.

Le composant tAzureStorageQueueInputLoop Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <code>Repository &gt; Metadata</code> seront réutilisées par ce composant. Vous devez cliquer sur le bouton <code>[...]</code> et, dans la boîte de dialogue <code>Repository Content</code>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste <code>Connection Component</code>.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage. |

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard `Storage accounts` du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations, |
consultez Utilisation des signatures d'accès partagé (SAP).

Dans le champ Azure Shared Access Signature qui s'affiche, saisissez votre signature d'accès partagé entre guillemets doubles. Vous pouvez obtenir l'URL de la signature d'accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l'URL est le suivant : https://<$storagename>.<$service>.core.windows.net/<$sastoken>, où <$storagename> est le nom du compte de stockage, <$service> le nom du service autorisé (blob, file, queue ou table) et <$sastoken> est la valeur du jeton de signature d'accès partagé. Pour plus d'informations, consultez Constructing the Account SAS URI (en anglais).

Notez que SAS a une période de validité, vous pouvez configurer l'heure de début de validité de SAS et son heure d'expiration, après laquelle SAS n'est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

Queue name
Spécifiez le nom de la file Azure de laquelle récupérer les messages.

Schema et Edit schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

- Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Le schéma de ce composant est prédéfini avec les colonnes suivantes :

- MessageId : ID du message.
- MessageContent : corps du message.
- InsertionTime : l'heure à laquelle le message a été ajouté à la file.
- ExpirationTime : l'heure à laquelle le message va expirer.
- NextVisibleTime : l'heure à laquelle le message sera à nouveau visible.
- DequeueCount : nombre de fois que le message a été retiré de la file. Cette valeur est incrémentée chaque fois que le message est retiré de la file, mais n'est pas incrémentée lorsque le message est regardé.
- PopReceipt : valeur de l'accusé réception pop requis pour supprimer le message.
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th><strong>Number of messages</strong></th>
<th>Saisissez le nombre de messages à récupérer de la file en une fois, jusqu’à 32.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Loop wait time</strong></td>
<td>Spécifiez (en secondes) la durée durant laquelle la boucle attend que le message arrive dans la file, avant de recommencer la boucle.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.</td>
</tr>
</tbody>
</table>

### Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th><strong>NB_LINE</strong></th>
<th>Nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>QUEUE_NAME</strong></td>
<td>Nom de la file Azure. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

Règle d’utilisation

Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie.
Scénario associé

Aucun scénario n’est disponible pour ce composant.
**tAzureStorageQueueList**

Ce composant retourne toutes les files associées au compte Azure Storage donné.

**Propriétés du tAzureStorageQueueList Standard**

Ces propriétés sont utilisées pour configurer le tAzureStorageQueueList s'exécutant dans le framework de Jobs Standard.

Le composant tAzureStorageQueueList Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Type</strong></td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage. |

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations, |
Dans le champ **Azure Shared Access Signature** qui s’affiche, saisissez votre signature d’accès partagé entre guillemets doubles. Vous pouvez obtenir l’URL de la signature d’accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l’URL est le suivant : `https://<$storagename>.<$service>.core.windows.net/$sastoken`, où `<$storagename>` est le nom du compte de stockage, `<$service>` le nom du service autorisé (`blob`, `file`, `queue` ou `table`) et `<$sastoken>` est la valeur du jeton de signature d’accès partagé. Pour plus d’informations, consultez **Constructing the Account SAS URI** (en anglais).

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.
- **Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Le schéma de ce composant est prédéfini et contient une colonne `QueueName` stockant le nom de chaque file à retourner.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Die on error

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.
**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métdonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>NUMBER_OF_QUEUES</th>
<th>Nombre de files retournées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie. |

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
**tAzureStorageQueueOutput**

Ce composant ajoute des messages à la fin d’une file Azure.
Notez que ce composant fonctionne uniquement avec Java 8.

**Propriétés du tAzureStorageQueueOutput Standard**

Ces propriétés sont utilisées pour configurer le composant tAzureStorageQueueOutput s’exécutant dans le framework de Jobs Standard.
Le composant tAzureStorageQueueOutput Standard appartient à la famille Cloud.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Type</strong></td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong>: les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong>: les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton […] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste **Connection Component**.

<table>
<thead>
<tr>
<th>Connection Component</th>
<th>Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Account Name</strong></td>
<td>Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Account Key</strong></th>
<th>Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Azure Shared Access Signature</strong></td>
<td>Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations,</td>
</tr>
</tbody>
</table>
Dans le champ **Azure Shared Access Signature** qui s’affiche, saisissez votre signature d’accès partagé entre guillemets doubles. Vous pouvez obtenir l’URL de la signature d’accès partagé pour chaque service autorisé sur le portail de Microsoft Azure, après génération de la signature. Le format de l’URL est le suivant: `https://<$storagename>.$<service>.core.windows.net/<$sastoken>`

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

### Queue name
Spécifiez le nom de la file Azure à laquelle ajouter les messages.

### Schema et Edit schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In**: Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Le schéma de ce composant est prédéfini et contient une colonne **MessageContent** stockant le corps de chaque message.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métdonnée du schéma dans la fenêtre [Repository Content].
Die on error

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

Advanced settings

tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

Global Variables

<table>
<thead>
<tr>
<th>NB_LINE</th>
<th>Nombre de messages traités. Cette variable est une variable <em>After</em> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_SUCCESS</td>
<td>Nombre de messages mis dans la file. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td>NB_REJECT</td>
<td>Nombre de messages rejetés. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Usage

Règle d’utilisation

Ce composant est généralement utilisé comme composant de fin dans un Job ou un sous-job et nécessite un lien d’entrée.

Scénario associé

Aucun scénario n’est disponible pour ce composant.
tAzureStorageQueuePurge

Ce composant purge les messages dans une file Azure.

Propriétés du tAzureStorageQueuePurge Standard

Ces propriétés sont utilisées pour configurer le composant tAzureStorageQueuePurge s’exécutant dans le framework de Jobs Standard.

Le composant tAzureStorageQueuePurge Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

| Property Type | Sélectionnez la manière de configurer les informations de connexion.
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste <strong>Connection Component</strong>.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez dans la liste déroulante le composant dont les informations de connexion seront utilisées pour configurer la connexion à Azure Storage. |

| Account Name | Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard Storage accounts du système Microsoft Azure Storage à utiliser. Assurez-vous que l’administrateur du système a donné les droits d’accès appropriés à ce compte de stockage. |

| Account Key | Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte. |

| Protocol | Sélectionnez le protocole de la connexion à créer. |

| Use Azure Shared Access Signature | Cochez cette case pour utiliser une signature d’accès partagé pour accéder aux ressources du stockage sans utiliser de clé de compte. Pour plus d’informations, |

Notez que SAS a une période de validité, vous pouvez configurer l’heure de début de validité de SAS et son heure d’expiration, après laquelle SAS n’est plus valide lors de la génération. Vous devez vous assurer que votre SAS est toujours valide lorsque vous exécutez votre Job.

<table>
<thead>
<tr>
<th><strong>Queue name</strong></th>
<th>Spécifiez le nom de la file Azure dans laquelle les messages seront purgés.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Die on error</strong></td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.</td>
</tr>
</tbody>
</table>

**Advanced settings**

- **tStatCatcher Statistics**
  Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Global Variables**

- **ERROR_MESSAGE**
  Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

**Utilisation**

- **Règle d’utilisation**
  Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job.

### Scénario associé

Aucun scénario n’est disponible pour ce composant.
tBarChart

Ce composant génère un diagramme en barres à partir des données d’entrée afin de faciliter l’analyse technique.

Le composant tBarChart lit des données à partir d’un flux d’entrée et transforme les données en diagramme en barres dans un fichier image au format PNG.

**Propriétés du tBarChart Standard**

Ces propriétés sont utilisées pour configurer le tBarChart s’exécutant dans le framework de Jobs Standard.

Le composant tBarChart Standard appartient à la famille Business Intelligence.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs. Cliquez sur `Edit schema` pour modifier le schéma. Si le schéma est en mode `Repository`, trois options sont disponibles :

  - **View schema** : sélectionnez cette option afin de voir le schéma.

  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur `No` et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

  **Remarque :**

  Le schéma du tBarChart contient trois colonnes en lecture seule nommées respectivement `series` (type String), `category` (type String), et `value` (type Integer), selon un ordre prédéterminé. Les données de toutes les autres colonnes seront simplement passées au composant suivant, s’il y en a un, mais elles ne seront pas présentes dans le diagramme en barres. |
### tBarChart

**Built-in**: Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

**Repository**: Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

**Sync columns**
Permet de synchroniser le schéma du fichier d'entrée avec le schéma du fichier de sortie. La fonction Sync ne s'affiche que lorsqu'une connexion de type *Row* est connectée au composant de sortie.

**Generated image path**
Nom et chemin d'accès du fichier image de sortie.

**Chart title**
Saisissez le titre du diagramme en barres à générer.

**Include legend**
Cochez cette case si vous souhaitez inclure une légende à votre diagramme en barres, afin d'afficher toutes les séries *(series)* dans des couleurs différentes.

**3Dimensions**
Cochez cette case afin de créer une image en trois dimensions. Par défaut, cette case est cochée et les barres représentant les séries *(series)* de chaque catégorie *(category)* sont empilées les unes sur les autres. Si cette case est décochée, une image en deux dimensions est créée, et les barres s'affichent les unes à côté des autres le long de l'axe *category*.

**Image width** et **Image height**
Saisissez la largeur et la hauteur du fichier image, exprimées en pixels.

**Category axis name** et **Value axis name**
Saisissez le nom de l'axe *category* et le nom de l'axe *value*.

**Foreground alpha**
Saisissez un entier entre 0 et 100 afin de définir la transparence de l'image. Plus ce nombre est petit, plus votre image sera transparente.

**Plot orientation**
Choisissez l'orientation qu'aura votre diagramme en barres: *VERTICAL* ou *HORIZONTAL*.

### Advanced settings

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

- **NB_LINE**: nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.
- **ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères.
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

---

**Utilisation**

| Règle d’utilisation | Ce composant est un composant de sortie. Par conséquent, il requiert un composant d’entrée ainsi qu’une connexion de type **Row Main**. |

---

**Scénario : Créer un diagramme en barres à partir de données d’entrée**

Ce scénario décrit un Job simple qui lit les données d’un fichier CSV et les transforme en diagramme en barres. Le fichier d’entrée est présenté ci-dessous :

| City;Population(x1000);LandArea(km2);PopulationDensity(people/km2) |
| Beijing;10233;1418;7620 |
| Moscow;10452;1081;9644 |
| Seoul;10422;605;17215 |
| Tokyo;8731;617;14151 |
| Jakarta;8490;664;12738 |
| New York;8310;789;10452 |

Le fichier d’entrée ayant une structure différente de celle requise pour l’utilisation d’un **tBarChart**, un **tMap** sera utilisé dans ce scénario, afin de mettre en correspondance les données avec un fichier CSV à trois colonnes, avant d’utiliser le **tBarChart** pour générer un fichier représentant le diagramme en barres.

**Remarque :**

Le **tMap** est généralement utilisé afin d’adapter le schéma d’entrée conformément à la structure du schéma du composant **tBarChart**. Pour plus d’informations concernant le composant **tMap**, consultez le **Guide utilisateur du Studio Talend** ainsi que **tMap** à la page 2113.

**Déposer et relier les composants**

**Procédure**

1. A partir de la **Palette**, déposez les composants suivants dans l’espace de modélisation graphique : deux composants **tFileInputDelimited**, un **tMap**, trois composants **tFileOutputDelimited** et un **tBarChart**. Renommez les composants comme vous le souhaitez pour faciliter le travail de conception.
2. Connectez le composant **tPrejob** à un **tFileDelete** à l'aide d'une connexion **Trigger > On Component Ok** et connectez le composant **tPostjob** au second **tFileDelete** via le même type de connexion.

3. Connectez le premier composant **tFileInputDelimited** au **tMap** à l'aide d'une connexion **Row > Main**.

4. Connectez le composant **tMap** au premier **tFileOutputDelimited** à l'aide d'une connexion **Row > Main** et appelez-la **Population**.

5. Répétez les étapes ci-dessus pour connecter le composant **tMap** aux deux autres **tFileOutputDelimited** à l'aide d'une connexion **Row > Main** et appelez-les **Area** et **Density**.

6. Connectez le **tFileInputDelimited** au composant **tBarChart** à l'aide d'un lien **Row > Main**.

7. Connectez le premier composant **tFileInputDelimited** au second **tFileInputDelimited** en utilisant une connexion **Trigger > OnSubjobOk**.

8. Renommez les composants de façon à décrire leur fonction.

**Résultats**

**Lire les données source**

**Procédure**

1. Double-cliquez sur le premier composant **tFileInputDelimited** afin d'afficher l'onglet **Basic settings** de sa vue **Component**.
2. Renseignez le chemin d’accès au fichier à traiter dans le champ **File name**.

3. Spécifiez dans le champ **Header** le nombre de lignes d’en-tête. Dans ce scénario, il n’y a qu’une seule ligne.

4. Cliquez sur **Edit schema** pour décrire la structure des données du fichier d’entrée. Dans ce scénario, le schéma d’entrée est composé de quatre colonnes : City, Population, Area, et Density. Une fois les noms de colonnes et les types de données définis, cliquez sur **OK** pour fermer la boîte de dialogue du schéma.

---

**Adapter les données source au schéma du tBarChart**

**Procédure**

1. Double-cliquez sur le **tMap** pour ouvrir l’éditeur **Map Editor**.

   Vous pouvez voir une table d’entrée dans le panneau d’entrée, row1 dans cet exemple et dans le panneau de sortie, trois tables de sortie vides appelées *Population, Area et Density*.

2. Utilisez l’éditeur **Schema editor** pour ajouter trois colonnes à chacune des tables de sortie : series (type String), category (type String) et value (type Integer).
3. Dans le champ **Expression** de chaque table de sortie, saisissez le texte à afficher dans la légende du graphique, **dans cet exemple** “Population (x1000 people)”, “Land area (km2)”, et “Population density (people/km2)”.  
4. Déposez la colonne **City** de la table d’entrée dans la colonne **category** de chaque table de sortie.  
5. Déposez la colonne **Population** de la table d’entrée dans la colonne **value** de la table **Population**.  
6. Déposez la colonne **Area** de la table d’entrée dans la colonne **value** de la table **Area**.  
7. Déposez la colonne **Density** de la table d’entrée dans la colonne **value** de la table **Density**.  

8. Cliquez sur **OK** pour sauvegarder les mappings, fermer l’éditeur et propager les schémas de sortie aux composants de sortie.

**Générer le fichier d’entrée temporaire**

**Procédure**

1. Double-cliquez sur le premier composant **tFileOutputDelimited** pour ouvrir sa vue **Basic settings**.

   ![Image]

2. Dans le champ **File Name**, définissez un fichier CSV dans lequel envoyer les flux de données mappés. Dans cet exemple, nommez le fichier de sortie à créer **Temp.csv**. Ce fichier sera utilisé comme fichier d’entrée pour le composant **tBarChart**.
3. Cochez la case **Append**.

4. Définissez les paramètres des deux autres composants *tFileOutputDelimited* exactement de la même manière que le premier *tFileOutputDelimited*.

**Remarque :**
Notez que l’ordre des flux de données sortant du composant *tMap* n’est pas nécessairement l’ordre dans lequel les données seront écrites dans le composant cible. Pour vous assurer que le fichier est correctement généré, supprimez, s’il existe, le fichier portant le même nom avant d’exécuter le Job. Vérifiez que la case **Append** est cochée dans tous les composants *tFileOutputDelimited* afin que tous les flux de données mappés aillent dans le même fichier, sans écraser les données existantes.

**Configurer les données d’entrée pour le tBarChart**

**Procédure**

1. Double-cliquez sur le second composant *tFileInputDelimited* afin d’afficher l’onglet **Basic settings** de sa vue **Component**.

2. Renseignez le champ *File name* avec le chemin d’accès au fichier, ainsi que son nom, comme défini dans chaque composant *tFileOutputDelimited*. Dans cet exemple, le fichier d’entrée est *LargeCities_mapped.csv*.

3. Double-cliquez sur le composant *tBarChart* pour afficher l’onglet **Basic settings**.
4. Dans le champ **Generated image path**, renseignez le chemin d'accès du fichier image à générer.

5. Dans le champ **Chart title**, saisissez un titre pour le diagramme en barres.

6. Saisissez un nom pour les axes **category** et **series**.

7. Si nécessaire, définissez la taille (la largeur dans le champ **Image Width**, la hauteur dans le champ **Image height**) et le degré de transparence de l’image (dans le champ **Foreground alpha**). Dans ce scénario, laissez les paramètres par défaut.

8. Cliquez sur **Edit schema** afin d’ouvrir la boîte de dialogue du schéma.

9. Copiez toutes les colonnes du schéma de sortie dans le schéma d’entrée en cliquant sur la double-flèche pointant vers la gauche.

   Cliquez sur **OK** pour fermer la boîte de dialogue du schéma.

**Supprimer le fichier temporaire**

**Pourquoi et quand exécuter cette tâche**

La fonction des composants **tPrejob** et **tPostjob** est de déclencher les sous-jobs connectés et n’ont pas besoin d’être configurés. Vous devez simplement définir les propriétés des deux composants **tFileDelete**.
**Procédure**

1. Double-cliquez sur le premier composant **tFileDelete** pour afficher sa vue **Basic settings**.

![Image of Del_Temp_Before(tFileDelete_1) with File Name set to "D:/Input/Temp.csv" and documentation note]

2. Dans le champ **File name**, renseignez le chemin d’accès du fichier temporaire.

Si la case **Fail on error** est cochée et que le sous-job de pré-traitement échoue à cause d’erreurs (par exemple le fichier à supprimer n’existe pas), l’échec empêche le lancement du sous-job principal. Dans cette situation, décochez la case **Fail on error** pour éviter cette interruption.

**Exécuter votre Job**

**Procédure**

1. Sauvegardez votre Job.

2. Appuyez sur **F6** pour l’exécuter.

Un diagramme en barres est généré selon les critères définis.
Comparison of large cities

Values

Cities

Beijing  Moscow  Seoul  Tokyo  Jakarta  New York

Legend:
- Population density (people/km²)
- Land area (km²)
- Population (×1000 people)
tBigQueryBulkExec

Ce composant transfère des données dans Google BigQuery.

Les composants tBigQueryOutputBulk et tBigQueryBulkExec sont généralement utilisés ensemble comme les deux étapes d'un processus. Dans la première étape, un fichier de sortie est généré. Dans la seconde étape, ce fichier est utilisé pour alimenter un ensemble de données. Ces deux étapes sont fusionnées dans le composant tBigQueryOutput, détaillé dans une section séparée. L'utilisation de deux composants séparés permet de transformer les données avant de les charger.

Ce composant transfère un fichier donné de Google Cloud Storage à Google BigQuery, ou charge un fichier donné dans Google Cloud Storage et le transfère dans Google BigQuery.

Propriétés du tBigQueryBulkExec Standard

Ces propriétés sont utilisées pour configurer le tBigQueryBulkExec s'exécutant dans le framework de Jobs Standard.

Le composant tBigQueryBulkExec Standard appartient à la famille Big Data.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
| | - View schema : sélectionnez cette option afin de voir le schéma.  
| | - Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
| | - Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |
| **Client ID et Client secret** | Collez l'ID du client et son mot de passe ("secret") créés et visibles dans l'onglet API Access du projet hébergeant les services BigQuery et Cloud Storage à utiliser.

Pour saisir le Secret du client, cliquez sur le bouton [...] à côté du champ Client Secret, puis, dans la boîte de dialogue qui s'ouvre, saisissez le Secret du client entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres. |
|-------------------------------|---|
| **Project ID** | Collez l'ID du projet hébergeant le service BigQuery à utiliser.

L'ID de votre projet se trouve dans l'URL de la console de l'API Google ou en passant votre curseur sur le nom du projet dans BigQuery Browser Tool. |
| **Authorization code** | Collez le code d'autorisation fourni par Google pour l'accès en cours de construction.

Pour obtenir le code d'autorisation, vous devez exécuter le Job utilisant ce composant. Lorsque l'exécution du Job est en pause pour afficher une URL, vous devez vous rendre à l'URL donnée afin de copier le code d'autorisation. |
| **Dataset** | Saisissez le nom de l'ensemble de données dans lequel vous devez transférer des données. |
| **Table** | Saisissez le nom de la table dans laquelle vous devez transférer les données.

Si la table n'existe pas, cochez la case **Create the table if it doesn't exist.** |
| **Action on data** | Sélectionnez l'action à effectuer dans la liste déroulante lors du transfert des données dans la table cible. Les actions disponibles sont :

- **Truncate** : vide le contenu de la table et la repeuple avec les données transférées.
- **Append** : ajoute des lignes aux données existant dans la table.
- **Empty** : peuple une table vide. |
| **Bulk file already exists in Google storage** | Cochez cette case pour réutiliser les informations d'authentification pour vous connecter à Google Cloud Storage, puis renseignez les champs **File** (fichier) et **Header** (en-tête). |
| **Access key et Secret key** | Collez les informations d'authentification obtenues de Google pour envoyer des requêtes à Google Cloud Storage.

Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres. |
Ces clés peuvent être consultées dans la vue Interoperable Access, sous l’onglet Google Cloud Storage du projet.

**File to upload**

Lorsque les données à transférer à BigQuery ne sont pas stockées dans Google Cloud Storage, parcourrez votre système ou saisissez le chemin d'accès à ces données.

**Bucket**

Saisissez le nom du bucket, du conteneur Google Cloud Storage, contenant les données à transférer à Google BigQuery.

**File**

Saisissez le répertoire des données stockées dans Google Cloud Storage et à transférer à BigQuery. Si les données ne sont pas dans Google Cloud Storage, ce répertoire est utilisé comme destination intermédiaire avant transfert des données à BigQuery.

**Header**

Configurez la valeur pour ignorer l’en-tête des données transférées. Par exemple, saisissez 0 afin de n’ignorer aucune ligne, si vos données ne contiennent pas d’en-tête.

**Die on error**

Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

### Advanced settings

**token properties File Name**

Saisissez le chemin d’accès ou parcourez votre système jusqu’au fichier de jeton à utiliser.

Lors de la première exécution du Job avec le code **Authorization code** de Google BigQuery, vous devez saisir dans ce champ le répertoire et le nom du fichier contenant le nouveau jeton (refresh token) à créer et à utiliser. Si ce fichier de jeton a été créé et que vous devez le réutiliser, vous devez spécifier son répertoire ainsi que le nom du fichier dans ce champ.

Si vous saisissez uniquement le nom du fichier de jeton, le **Studio Talend** considère que le répertoire de ce fichier est la racine du dossier du **Studio Talend**.

Pour plus d’informations concernant le refresh token, consultez le manuel de Google BigQuery.

**Set the field delimiter**

Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.

**Encoding**

 Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
### tStatCatcher Statistics

Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

#### ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

#### Règle d’utilisation

Ce composant est utilisable en tant que composant standalone.

### Scénario associé

Pour un scénario associé, consultez Scénario : Ecrire des données dans BigQuery à la page 377.
**tBigQueryInput**

Ce composant effectue les requêtes supportées par Google BigQuery.

Le composant tBigQueryInput se connecte à Google BigQuery et y effectue des requêtes.

**Propriétés du tBigQueryInput Standard**

Ces propriétés sont utilisées pour configurer le tBigQueryInput s'exécutant dans le framework de Jobs Standard.

Le composant tBigQueryInput Standard appartient à la famille Big Data.

Le composant de ce framework est toujours disponible.

**Basic settings**

| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

| **Built-In** | Le schéma est créé et conservé ponctuellement pour ce composant uniquement. Voir également le Guide utilisateur du Studio Talend.

| **Repository** | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

| **Client ID et Client secret** | Collez l’ID du client et son mot de passe ("secret") créés et visibles dans l’onglet API Access du projet hébergeant les services BigQuery et Cloud Storage à utiliser.
Pour saisir le Secret du client, cliquez sur le bouton [...] à côté du champ **Client Secret**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le Secret du client. |
entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.

| **Project ID** | Collez l’ID du projet hébergeant le service BigQuery à utiliser.  
L’ID de votre projet se trouve dans l’URL de la console de l’API Google ou en passant votre curseur sur le nom du projet dans BigQuery Browser Tool. |
| **Authorization code** | Collez le code d’autorisation fourni par Google pour l’accès en cours de construction.  
Pour obtenir le code d’autorisation, vous devez exécuter le Job utilisant ce composant. Lorsque l’exécution du Job est en pause pour afficher une URL, vous devez vous rendre à l’URL donnée afin de copier le code d’autorisation. |
| **Query** | Saisissez la requête à utiliser. |
| **Result size** | Sélectionnez l’option en fonction du volume du résultat de la requête.  
Par défaut, l’option **Small** est utilisée, mais, lorsque le résultat de requête est plus grand que la taille de réponse maximale (maximum response size, en anglais), vous devez sélectionner l’option **Large**.  
Si le volume du résultat n’est pas certain, sélectionnez **Auto**. |

### Advanced settings

| **token properties File Name** | Saisissez le chemin d’accès ou parcourez votre système jusqu’au fichier de jeton à utiliser.  
Lors de la première exécution du Job avec le code **Authorization code** de Google BigQuery, vous devez saisir dans ce champ le répertoire et le nom du fichier contenant le nouveau jeton (refresh token) à créer et à utiliser. Si ce fichier de jeton a été créé et que vous devez le réutiliser, vous devez spécifier son répertoire ainsi que le nom du fichier dans ce champ.  
Si vous saisissez uniquement le nom du fichier de jeton, le Studio Talend considère que le répertoire de ce fichier est la racine du dossier du Studio Talend.  
Pour plus d’informations concernant le refresh token, consultez le manuel de Google BigQuery. |
| **Advanced Separator (for number)** | Cochez cette case pour modifier le séparateur utilisé pour les nombres. |
| **Encoding** | Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données. |
Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

Global Variables

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Ce composant est utilisé en tant que composant d’entrée. Il envoie les données extraites au composant suivant.

Scénario : Exécuter une requête dans BigQuery

Ce scénario utilise deux composants pour effectuer une requête SELECT dans BigQuery et afficher les résultats dans le Studio Talend.

La capture d’écran suivante présente le schéma de la table UScustomer utilisée comme exemple pour exécuter la requête SELECT.
L'objectif est de sélectionner les enregistrements contenant des États (des Etats-Unis) et de compter le nombre d'occurrences de chaque État dans ces enregistrements.

**Relier les composants**

**Procédure**

1. Dans la perspective **Integration** du **Studio Talend**, créez un Job vide, nommé, par exemple, **BigQueryInput**, depuis le nœud **Job Designs** de la vue **Repository**.
   
   Pour plus d'informations concernant la création d'un Job, consultez le Guide utilisateur du **Studio Talend**.

2. Déposez un composant **tBigQueryInput** et un **tLogRow** dans l'espace de modélisation graphique.

3. Connectez-les à l'aide d'un lien **Row > Main**.

**Créer la requête**

**Créer l'accès à BigQuery**

**Procédure**

1. Double-cliquez sur le **tBigQueryInput** pour ouvrir sa vue **Component**.
2. Cliquez sur **Edit schema** pour ouvrir l’éditeur

![Edit schema](image)

3. Cliquez deux fois sur le bouton pour ajouter deux lignes et saisissez le nom de ces lignes dans la colonne **Column**. Dans ce scénario, nommez-les : **States** et **Count**.

4. Cliquez sur **OK** afin de valider les modifications et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

5. Dans votre navigateur Web, rendez-vous sur la page de la console Google APIs afin d’accéder au projet Google hébergeant les services BigQuery et Cloud Storage à utiliser.


7. Dans la vue **Component** du **Studio Talend**, collez, dans les champs correspondants, l’ID Client, le mot de passe Client et l’ID du projet, disponibles dans la vue API Access.

**Obtenir le code d’autorisation**

**Procédure**

1. Dans la vue **Run** du **Studio Talend**, cliquez sur **Run** pour exécuter le Job. L’exécution se met en pause à un moment, pour afficher dans la console l’URL à utiliser pour obtenir le code d’autorisation.

2. Rendez-vous à l’URL, dans votre navigateur Web et copiez le code d’autorisation affiché.

3. Dans la vue **Component** du composant **tBigQueryInput**, collez le code dans le champ **Authorization Code**.
Écrire la requête

**Procédure**

Dans le champ **Query**, saisissez

```
select States, count(*) as Count from documentation.UScustomer group by States
```

**Exécuter le Job**

**Pourquoi et quand exécuter cette tâche**

Le composant tLogRow présente les résultats d’exécution du Job. Vous pouvez configurer le mode de présentation dans le vue **Component**.

Pour ce faire, double-cliquez sur le tLogRow afin d’ouvrir sa vue **Component**. Dans la zone **Mode**, sélectionnez l’option **Table (print values in cells of a table)**.

**Procédure**

Pour exécuter le Job, appuyez sur **F6**.

**Résultats**

La vue **Run** s’ouvre automatiquement et vous pouvez visualiser les résultats de l’exécution.

![Screenshot du Job BigQuerySelect](image-url)
tBigQueryOutput

Ce composant transfère les données fournies par le composant précédent dans Google BigQuery.
Le tBigQueryOutput écrit les données qu'il reçoit dans un répertoire spécifié par l'utilisateur et transfère les données dans Google BigQuery via Google Cloud Storage.

Propriétés du tBigQueryOutput Standard

Ces propriétés sont utilisées pour configurer le tBigQueryOutput s'exécutant dans le framework de Jobs Standard.
Le composant tBigQueryOutput Standard appartient à la famille Big Data.
Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Local filename</td>
<td>Parcourez votre système ou saisissez le chemin d'accès au fichier dans lequel vous souhaitez écrire les données reçues.</td>
</tr>
<tr>
<td>Append</td>
<td>Cochez cette case pour ajouter des lignes aux données existant dans le fichier spécifié dans le champ <strong>Local filename</strong>.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>
| **Client ID et Client secret** | Collez l'ID du client et son mot de passe ("secret") créés et visibles dans l'onglet API Access du projet hébergeant les services BigQuery et Cloud Storage à utiliser.  
Pour saisir le Secret du client, cliquez sur le bouton [...] à côté du champ **Client Secret**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le Secret du client entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Project ID** | Collez l'ID du projet hébergeant le service BigQuery à utiliser.  
L'ID de votre projet se trouve dans l'URL de la console de l'API Google ou en passant votre curseur sur le nom du projet dans BigQuery Browser Tool. |
| **Authorization code** | Collez le code d'autorisation fourni par Google pour l'accès en cours de construction.  
Pour obtenir le code d'autorisation, vous devez exécuter le Job utilisant ce composant. Lorsque l'exécution du Job est en pause pour afficher une URL, vous devez vous rendre à l'URL donnée afin de copier le code d'autorisation. |
| **Dataset** | Saisissez le nom de l'ensemble de données dans lequel vous souhaitez transférer des données. |
| **Table** | Saisissez le nom de la table dans laquelle vous devez transférer les données.  
Si la table n'existe pas, cochez la case **Create the table if it doesn't exist**. |
| **Action on data** | Sélectionnez l'action à effectuer dans la liste déroulante lors du transfert des données dans la table cible. Les actions disponibles sont :  
• **Truncate** : vide le contenu de la table et la repeuple avec les données transférées.  
• **Append** : ajoute des lignes aux données existant dans la table.  
• **Empty** : alimente une table vide. |
| **Access key et Secret key** | Collez les informations d'authentification obtenues de Google pour envoyer des requêtes à Google Cloud Storage.  
Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. |
Ces clés peuvent être consultées dans la vue Interoperable Access, sous l’onglet Google Cloud Storage du projet.

**Bucket**
Saisissez le nom du bucket, du conteneur Google Cloud Storage, contenant les données à transférer à Google BigQuery.

**File**
Saisissez le répertoire des données stockées dans Google Cloud Storage et à transférer à BigQuery.
Si les données ne sont pas dans Google Cloud Storage, ce répertoire est utilisé comme destination intermédiaire avant transfert des données à BigQuery.
Notez que le nom du répertoire doit être identique au nom de fichier spécifié dans le champ *Local filename*.

**Header**
Configurez la valeur pour ignorer l’en-tête des données transférées. Par exemple, saisissez 0 afin de n’ignorer aucune ligne, si vos données ne contiennent pas d’en-tête.

**Die on error**
Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

**Advanced settings**

| **token properties File Name** | Saisissez le chemin d’accès ou parcourez votre système jusqu’au fichier de jeton à utiliser.

Lors de la première exécution du Job avec le code *Authorization code* de Google BigQuery, vous devez saisir dans ce champ le répertoire et le nom du fichier contenant le nouveau jeton (refresh token) à créer et à utiliser. Si ce fichier de jeton a été créé et que vous devez le réutiliser, vous devez spécifier son répertoire ainsi que le nom du fichier dans ce champ.

Si vous saisissez uniquement le nom du fichier de jeton, le *Studio Talend* considère que le répertoire de ce fichier est la racine du dossier du *Studio Talend*.

Pour plus d’informations concernant le refresh token, consultez le manuel de Google BigQuery. |
| **Field Separator** | Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées. |
| **Create directory if not exists** | Cochez cette case pour créer le répertoire que vous avez défini dans le champ *File* pour Google Cloud Storage, s’il n’existe pas. |
| **Custom the flush buffer size** | Saisissez le nombre de lignes à traiter avant de libérer la mémoire. |
| **Check disk space** | Cochez cette case pour retourner une exception durant l’exécution si le disque est plein. |
**Encoding**
Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.
- Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.
- Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
- Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**
Ce composant est un composant de sortie. Il reçoit les données du composant précédent, comme le **tFileInputDelimited**, le **tMap** ou le **tMysqlInput**.

### Scénario : Ecrire des données dans BigQuery

Ce scénario utilise deux composants pour écrire des données dans Google BigQuery.

### Relier les composants

**Procédure**

1. Dans la perspective **Integration** du **Studio Talend**, créez un Job vide, nommé par exemple **WriteBigQuery**, depuis le nœud **Job Designs** de la vue **Repository**.
Pour plus d’informations concernant la création d’un Job, consultez le Guide utilisateur du Studio Talend.

2. Déposez un tRowGenerator et un tBigQueryOutput dans l’espace de modélisation graphique.
   Le composant tRowGenerator génère les données à transférer dans Google BigQuery, dans ce scénario. Dans un cas d’utilisation réelle, vous pouvez utiliser d’autres composants, comme le tMysqlInput ou le tMap à la place du tRowGenerator, afin de créer un processus plus complexe pour préparer vos données à transférer.

3. Connectez les composants à l’aide d’un lien Row > Main.

**Préparer les données à transférer**

**Procédure**

1. Double-cliquez sur le tRowGenerator pour ouvrir sa vue Component.

2. Cliquez sur RowGenerator Editor pour ouvrir l’éditeur.

3. Cliquez trois fois sur le bouton pour ajouter trois lignes à la table Schema.

4. Dans la colonne Column, saisissez le nom de votre choix pour chacune des lignes. Par exemple, nommez les lignes fname, lname et States.

5. Dans la colonne Functions, sélectionnez TalendDataGenerator.getFirstName pour la ligne fname, TalendDataGenerator.getLastName pour lname et TalendDataGenerator.getUsState pour la ligne States.

6. Dans le champ Number of Rows for RowGenerator, saisissez, 100, par exemple, afin de définir le nombre de lignes à générer.

7. Cliquez sur OK pour valider les modifications.
Configurer l'accès à BigQuery et Cloud Storage

 Créer l'accès à BigQuery

Procédure

1. Double-cliquez sur le `tBigQueryOutput` pour ouvrir sa vue `Component`.

2. Cliquez sur **Sync columns** afin de récupérer le schéma du composant précédent.

3. Dans le champ **Local filename**, saisissez le chemin d'accès au répertoire dans lequel vous souhaitez créer le fichier à transférer dans BigQuery.

4. Dans votre navigateur Web, rendez-vous sur la page de la console Google APIs afin d'accéder au projet Google hébergeant les services BigQuery et Cloud Storage à utiliser.

5. Cliquez sur l'onglet API Access pour ouvrir la vue correspondante.

6. Dans la vue **Component** du `Studio Talend`, collez, dans les champs correspondants, l'ID Client, le mot de passe Client et l'ID du projet, disponibles dans la vue API Access.

7. Dans le champ **Dataset**, saisissez le nom de l'ensemble de données dans lequel vous souhaitez transférer des données. Dans ce scénario, il est nommé `documentation`.

L'ensemble de données doit exister dans BigQuery. La capture d'écran suivante présente l'ensemble de données utilisé dans ce scénario.
Dans le champ **Table**, saisissez le nom de la table dans laquelle vous souhaitez écrire les données, par exemple, *UScustomer*. Si la table n’existe pas dans votre BigQuery, sélectionnez **Create the table if it doesn’t exist**.

9. Dans le champ **Action on data**, sélectionnez l’action. Dans cet exemple, sélectionnez **Truncate** afin de vider le contenu, s’il y en a un, de la table cible et de la peupler à nouveau avec les données transférées.

### Créer l’accès à Cloud Storage

**Procédure**

1. Dans votre navigateur Web, rendez-vous sur la page de la console Google APIs afin d’accéder au projet Google hébergeant les services BigQuery et Cloud Storage à utiliser.

2. Cliquez sur **Google Cloud Storage > Interoperable Access** pour ouvrir la vue correspondante.

3. Dans la vue **Component** du *Studio Talend*, collez, dans les champs correspondants, la clé Access, le mot de passe Access, disponibles dans la vue Interoperable Access.


5. Dans le champ **File**, saisissez le chemin d’accès au répertoire de Google Cloud Storage dans lequel recevoir et créer le fichier à transférer dans BigQuery. Dans ce scénario, saisissez *gs://talend/documentation/biqury_UScustomer.csv*. Le nom du fichier doit être le même que celui défini dans le champ **Local filename**.

   Dépannage : si vous rencontrez des problèmes comme *Unable to read source URI* concernant le fichier stocké dans Google Cloud Storage, vérifiez que le nom du fichier est bien le même dans les deux champs.

6. Saisissez *0* dans le champ **Header** afin de prendre toutes les lignes en compte dans les données transférées.
Obtenir le code d’autorisation

Procédure

1. Dans la vue Run du Studio Talend, cliquez sur Run pour exécuter le Job. L’exécution se met en pause à un moment, pour afficher dans la console l’URL à utiliser pour obtenir le code d’autorisation.

2. Rendez-vous à l’URL, dans votre navigateur Web et copiez le code d’autorisation affiché.

3. Dans la vue Component du composant tBigQueryOutput, collez le code dans le champ Authorization Code.

Exécuter le Job

Procédure

Appuyez sur F6 pour exécuter le Job.

Résultats

La vue Run s’ouvre automatiquement et vous pouvez visualiser les résultats de l’exécution.

Les données sont transférées dans Google BigQuery.
### Query Results

<table>
<thead>
<tr>
<th>Row</th>
<th>name</th>
<th>lname</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Richard</td>
<td>Carter</td>
<td>Delaware</td>
</tr>
<tr>
<td>2</td>
<td>Warren</td>
<td>Truman</td>
<td>New York</td>
</tr>
<tr>
<td>3</td>
<td>Theodore</td>
<td>Wilson</td>
<td>Maryland</td>
</tr>
<tr>
<td>4</td>
<td>Andrew</td>
<td>Coolidge</td>
<td>Alaska</td>
</tr>
<tr>
<td>5</td>
<td>Ronald</td>
<td>Washington</td>
<td>Mississippi</td>
</tr>
</tbody>
</table>

**Query Results** 10:07am, 3 Aug 2012  
Download as CSV  Save as Table
tBigQueryOutputBulk

Ce composant crée un fichier .txt ou .csv pour des données volumineuses, afin que vous puissiez le traiter selon vos besoins, avant de le transférer dans Google BigQuery.

Les composants tBigQueryOutputBulk et tBigQueryBulkExec sont généralement utilisés ensemble comme les deux étapes d’un processus. Dans la première étape, un fichier de sortie est généré. Dans la seconde étape, ce fichier est utilisé pour alimenter un ensemble de données. Ces deux étapes sont fusionnées dans le composant tBigQueryOutput, détaillé dans une section séparée. L’utilisation de deux composants séparés permet de transformer les données avant de les charger.

Le composant tBigQuery écrit des données dans un fichier .txt ou .csv, prêt à être transféré dans Google BigQuery.

Propriétés du tBigQueryOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tBigQueryOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tBigQueryOutputBulk Standard appartient à la famille Big Data.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File name</th>
<th>Parcourez votre système ou saisissez le chemin d’accès au fichier .txt ou .csv à générer.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append</td>
<td>Cochez cette case pour écrire de nouvelles données à la suite des données existantes. Sinon, les données seront écrasées.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Field Separator</th>
<th>Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create directory if not exists</td>
<td>Cochez cette case pour créer le répertoire que vous avez défini dans le champ File pour Google Cloud Storage, s'il n'existe pas.</td>
</tr>
<tr>
<td>Custom the flush buffer size</td>
<td>Saisissez le nombre de lignes à traiter avant de libérer la mémoire.</td>
</tr>
<tr>
<td>Check disk space</td>
<td>Cochez cette case pour retourner une exception durant l’exécution si le disque est plein.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
</tbody>
</table>
Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d'utilisation | Ce composant est un composant de sortie. Il nécessite les données fournies par le composant précédent. |

**Scénario associé**

Pour un scénario associé, consultez Scénario : Ecrire des données dans BigQuery à la page 377.
tBonitaDeploy

Ce composant déploie un processus spécifique Bonita vers Bonita Runtime.

Le tBonitaDeploy configure un moteur Bonita Runtime et déploie un processus spécifique Bonita (un fichier .bar exporté de la solution Bonita solution) vers ce moteur.

**Propriétés du tBonitaDeploy Standard**

Ces propriétés sont utilisées pour configurer le tBonitaDeploy s'exécutant dans le framework de Jobs Standard.

Le composant tBonitaDeploy Standard appartient à la famille Business.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bonita version</td>
<td>Sélectionnez un numéro de version pour le moteur Bonita Runtime.</td>
</tr>
<tr>
<td>Bonita Runtime Environment File</td>
<td>Parcourez votre répertoire, ou saisissez le chemin d'accès au fichier d'environnement Bonita Runtime.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Ce champ est affiché uniquement lorsque vous sélectionnez la version 5.3.1 de Bonita dans la liste Bonita version.</td>
</tr>
<tr>
<td>Bonita Runtime Home</td>
<td>Parcourez votre système jusqu’au répertoire d’environnement de Bonita Runtime, ou saisissez son chemin d'accès.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Ce champ est affiché uniquement lorsque vous sélectionnez la version 5.6.1 de Bonita dans la liste Bonita version.</td>
</tr>
<tr>
<td>Bonita Runtime Jass File</td>
<td>Parcourez votre répertoire, ou saisissez le chemin d'accès au fichier jass Bonita Runtime.</td>
</tr>
<tr>
<td>Bonita Runtime logging file</td>
<td>Parcourez votre répertoire, ou saisissez le chemin d'accès au fichier de log Bonita Runtime.</td>
</tr>
<tr>
<td>Login Module</td>
<td>Saisissez le nom du module de login au moteur Bonita Runtime défini dans le fichier Bonita Runtime jaas.</td>
</tr>
<tr>
<td>Business Archive</td>
<td>Parcourez votre répertoire, ou saisissez le chemin d'accès au fichier .bar Bonita de traitement que vous souhaitez utiliser.</td>
</tr>
<tr>
<td>User name</td>
<td>Saisissez votre identifiant de connexion à Bonita.</td>
</tr>
<tr>
<td>Password</td>
<td>Saisissez votre mot de passe de connexion à Bonita.</td>
</tr>
</tbody>
</table>
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Die on error**

Cette case n’est pas cochée par défaut, pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur.

### Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les métagonées de traitement du Job au niveau du Job ainsi qu’au niveau de chaque composant.

### Global Variables

**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ProcessDefinitionUUID</strong></td>
<td>Identifiant du processus d’employé. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option. Une variable <em>Flow</em> fonctionne durant l’exécution d’un composant. Une variable <em>After</em> fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé en standalone. Pour utiliser ce composant, vous devez télécharger manuellement la solution Bonita à utiliser.

**Connections**

Liens de sortie (de ce composant à un autre):

| Trigger | Run if, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError |

Liens d’entrée (d’un autre composant à celui-ci):

| Trigger | Run if, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError |
### Limitation

Le fichier d'environnement Bonita Runtime, le fichier jaas Bonita Runtime et le fichier de log Bonita Runtime doivent être stockés sur le serveur d'exécution du Job, à l'aide de ce composant.

### Scénario associé

Pour un scénario associé, consultez *Scénario 1 : Exécuter un processus Bonita via un Job Talend* à la page 392.
**tBonitaInstantiateProcess**

Ce composant commence l’instance d’un processus spécifique déployé dans le moteur Bonita Runtime.

Le tBonitaInstantiateProcess instancie un processus existant dans un moteur Bonita Runtime puis lance l’exécution.

**Propriétés du tBonitaInstantiateProcess Standard**

Ces propriétés sont utilisées pour configurer le tBonitaInstantiateProcess s’exécutant dans le framework de Jobs Standard.

Le composant tBonitaInstantiateProcess Standard appartient à la famille Business.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.
Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur **Edit schema** afin de visualiser le schéma.
Pour ce composant, le schéma dépend du **Module** sélectionné.

**Remarque :**
La colonne `ProcessInstanceUUID` est prédéfinie dans le schéma de ce composant et réservée à l’identifiant de l’instance du processus créé.

| Bonita Client Mode | Sélectionnez le mode client que vous souhaitez utiliser pour instancier un processus Bonita.
Pour plus d’informations concernant les modes client de Bonita, consultez les manuels de Bonita.

Ce champ est disponible uniquement en mode HTTP client.

| Auth Username et Auth Password | Saisissez les informations d’authentification pour vous connecter au serveur d’application Web Bonita en tant qu’utilisateur technique.
Pour saisir le mot de passe, cliquez sur le bouton `[...]` à côté du champ `Password`, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
<table>
<thead>
<tr>
<th>Les informations par défaut sont automatiquement renseignées dans ces champ. Pour plus d'informations, consultez les manuels de Bonita. Ces champs sont disponibles uniquement en mode HTTP client.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bonita version</strong></td>
</tr>
<tr>
<td><strong>Bonita Runtime Environment File</strong></td>
</tr>
<tr>
<td><strong>Bonita Runtime Home</strong></td>
</tr>
<tr>
<td><strong>Bonita Runtime Jass File</strong></td>
</tr>
<tr>
<td><strong>Bonita Runtime logging file</strong></td>
</tr>
<tr>
<td><strong>Use Process ID</strong></td>
</tr>
</tbody>
</table>
Ce champ est disponible pour les modes Java client et HTTP client.

**User name**
Saisissez l’identifiant utilisé pour instancier ce processus.
Ce champ est disponible pour les modes Java client et HTTP client.

**Password**
Saisissez le mot de passe utilisé pour instancier ce processus.
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.
Ce champ est disponible uniquement en mode Java client.

**Die on error**
Cette case n’est pas cochée par défaut, pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur.

---

**Advanced settings**

**tStatCatcher Statistics**
Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu’au niveau de chaque composant.

---

**Global Variables**

**Global Variables**


- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est généralement utilisé en standalone ou en tant que composant de sortie. Pour utiliser ce composant, vous devez télécharger manuellement la solution Bonita à utiliser.</th>
</tr>
</thead>
</table>

| Connections | Liens de sortie (de ce composant à un autre) :  
Row : Main (fournit les paramètres de sortie à partir du processus).  
Liens d’entrée (d’un autre composant à celui-ci) :  
Row : Main.  
Trigger : Run if, OnComponentOk, OnComponentError, OnSubjobOk, OnSubjobError  
Pour plus d’informations à propos des connexions, consultez la section concernant les types de connexions dans le Guide utilisateur du Studio Talend. |

| Limitation | Le fichier d’environnement Bonita Runtime, le fichier jaas Bonita Runtime et le fichier de log Bonita Runtime doivent être stockés sur le serveur d’exécution du Job utilisant ce composant. |

**Scénario 1 : Exécuter un processus Bonita via un Job Talend**

Ce scénario décrit un Job déployant un processus Bonita dans le moteur Bonita Runtime et exécutant ce processus, dans lequel une requête sur le personnel est traitée.

Ce Job utilise trois composants.
- Un **tBonitaDeploy** : ce composant déploie un processus Bonita dans le moteur Bonita Runtime.
- Un **tFixedFlowInput** : ce composant génère le schéma utilisé comme paramètre d’exécution du processus déployé.
- Un **tBonitaInstantiateProcess** : ce composant exécute le processus déployé.
Avant de commencer à reproduire ce schéma, préparez votre fichier Bonita .bar. Vous devez exporter manuellement ce fichier depuis le système Bonita, puis le déployer dans le moteur Bonita Runtime, à l’aide, par exemple du composant **tBonitaDeploy**, présenté dans ce scénario. Dans ce scénario, ce fichier est **TEST--4.0.bar**. Une fois déployé, ce processus peut être vérifié via l’interface Bonita.

Pour reproduire ce scénario, procédez comme suit :

**Configurer le Job**

**Procédure**

1. Déposez les composants **tBonitaDeploy**, **tFixedFlowInput** et **tBonitaInstantiateProcess** de la Palette dans l’espace de modélisation graphique.
2. Cliquez-droit sur le composant **tBonitaDeploy** pour ouvrir son menu contextuel.
3. Sélectionnez **Trigger > OnSubjobOk** pour relier le **tBonitaDeploy** au **tFixedFlowInput**.
4. Cliquez-droit sur le **tFixedFlowInput** pour ouvrir son menu contextuel et sélectionnez **Row > Main** pour relier ce composant au **tBonitaInstantiateProcess**, à l’aide d’un lien **Main**.

**Configurer le déploiement du processus**

**Pourquoi et quand exécuter cette tâche**

Pour reproduire ce scénario, procédez comme suit :

**Procédure**

1. Double-cliquez sur le **tBonitaDeploy** afin d’ouvrir sa vue **Basic settings**.
2. Sélectionnez la version 5.3.1 de Bonita dans la liste **Bonita version**. La version sélectionnée doit être la même que celle du moteur Bonita Runtime que vous utilisez.


Pour les utilisateurs de la version 5.2.3 de Bonita, seuls les champs **Bonita Runtime Jaas File** et **Bonita Runtime Logging File** sont à renseigner.

Pour les utilisateurs de la version 5.6.1 de Bonita, parcourez votre système afin de renseigner le champ **Bonita Runtime Home** avec le répertoire d'environnement de Bonita Runtime.

4. Dans le champ **Business Archive**, parcourez votre répertoire jusqu'au fichier Bonita .bar, qui est le processus exporté de votre système Bonita et sera déployé dans le moteur Bonita Runtime.

5. Dans les champs **Username** et **Password**, saisissez vos informations de connexion à Bonita.

**Configurer le flux d'entrée**

**Procédure**

1. Double-cliquez sur le composant **tFixedFlowInput** afin d'ouvrir sa vue **Basic settings**.
2. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

3. Cliquez sur le bouton [+] pour ajouter une ligne. Nommez-la **Name**.

   Le nom est identique à celui du paramètre configuré dans Bonita, afin d'exécuter le même processus. Ainsi, Bonita reconnaît la colonne comme un paramètre valide et lit sa valeur pour instancier le processus.

4. Cliquez sur **OK**.

5. Dans la zone **Mode** de la vue **Basic settings**, sélectionnez l'option **Use inline table** puis cliquez sur le bouton [+] afin d'ajouter une ligne.

6. Dans le tableau, cliquez sur la ligne ajoutée et saisissez entre guillemets le nom de la personne de votre personnel : *ychen*, dont la requête sera traitée par le processus déployé.

**Configurer les paramètres du tBonitaInstantiateProcess**

**Procédure**

1. Double-cliquez sur le composant **tBonitaInstantiateProcess** afin d'ouvrir sa vue **Basic settings**.
2. Sélectionnez la version 5.3.1 de Bonita dans la liste **Bonita version**. La version sélectionnée doit être la même que celle du moteur Bonita Runtime que vous utilisez.


Pour les utilisateurs de la version 5.2.3 de Bonita, seuls les champs **Bonita Runtime Jaas File** et **Bonita Runtime Logging File** sont à renseigner.

Pour les utilisateurs de la version 5.6.1 de Bonita, parcourez votre système afin de renseigner le champ **Bonita Runtime Home** avec le répertoire d’environnement de Bonita Runtime.
4. Cochez la case Use Process ID pour activer le champ Process Definition Id.

5. Dans le champ Process Definition Id, cliquez entre les guillemets et appuyez sur Ctrl+Espace afin d’ouvrir la liste d’autocomplétion contenant les variables globales disponibles pour ce Job.

6. Double-cliquez sur la variable que vous souhaitez utiliser pour l’ajouter entre les guillemets.
Dans ce scénario, double-cliquez sur tBonitaDeploy_1_ProcessDefinitionUUID, qui récupère l’ID de définition du processus déployé par le composant tBonitaDeploy.

Remarque :

7. Dans les champs Username et Password, saisissez respectivement votre identifiant et votre mot de passe de connexion à Bonita.

**Exécuter le Job**

**Procédure**

Appuyez sur F6 pour exécuter le Job.

**Résultats**

Le processus est déployé dans le moteur Bonita Runtime et une instance est créée pour les requêtes concernant le personnel.
Scénario 2 : Ecrire en sortie l’UUID de l’instance du processus via le lien Row > Main

Ce scénario déploie un processus Bonita dans Bonita Runtime, démarre une instance et écrit en sortie l’UUID de l’instance du processus via le lien Row > Main.

Relier les composants

Procédure

1. Déposez de la Palette dans l’espace de modélisation graphique un composant tBonitaDeploy, un tBonitaInstantiateProcess et un tLogRow.
2. Renommez le tBonitaDeploy en deploy_process, le tBonitaInstantiateProcess en start_instance et le tLogRow en show_instance_uuid.
4. Reliez le tBonitaInstantiateProcess au tLogRow à l’aide d’un lien Row > Main.

Configurer les composants

Procédure

1. Double-cliquez sur le composant tBonitaDeploy pour ouvrir sa vue Basic settings.

2. Dans le champ Bonita Runtime Jaas File, spécifiez le chemin d’accès et le nom du fichier jaas.
   Dans le champ Bonita Runtime Logging File, spécifiez le chemin d’accès et le nom du fichier de log.
   Dans le champ Business Archive, spécifiez le chemin d’accès et le nom du processus Bonita.
3. Dans les champs Username et Password, saisissez vos informations d’authentification.
5. Dans le champ Bonita Runtime Jaas File, spécifiez le chemin d'accès et le nom du fichier jaas.

Dans le champ Bonita Runtime Logging File, spécifiez le chemin d'accès et le nom du fichier de log.


7. Dans les champs Username et Password, saisissez vos informations d'authentification.

8. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings.

9. Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour obtenir un meilleur affichage.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl+S afin de sauvegarder votre Job.

2. Appuyez sur F6 pour exécuter le Job.

```plaintext
**** Instance Bonita_V523_Process--1.0--6 created ****
__________show_instance_uuid__________
<table>
<thead>
<tr>
<th>ProcessInstanceUUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bonita_V523_Process--1.0--6</td>
</tr>
</tbody>
</table>
__________stat[i4istics] disconnected
```

Comme affiché ci-dessus, l’instance est créée et l’UUID est écrit en sortie.
tBoxConnection

Ce composant crée une connexion à Box que les autres composants Box peuvent réutiliser.
Ce composant crée une connexion à un compte Box donné.

Propriétés du tBoxConnection Standard

Ces propriétés sont utilisées pour configurer le tBoxConnection s’exécutant dans le framework de Jobs Standard.
Le composant tBoxConnection Standard appartient à la famille Cloud.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Key</td>
<td>Saisissez la clé client requise par Box afin d’accéder à l’API de Box. Afin d’obtenir la clé client et le Secret du client, vous devez créer un compte sur <a href="https://developers.box.com/">https://developers.box.com/</a> puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l’application du compte.</td>
</tr>
<tr>
<td>Access token</td>
<td>Renseignez le jeton d’accès requis par Box afin d’accéder au compte Box et le faire fonctionner. Pour plus d’informations concernant l’obtention du jeton d’accès et du jeton de rafraîchissement, consultez la documentation de Box disponible sur <a href="https://developers.box.com/">https://developers.box.com/</a> (en anglais).</td>
</tr>
<tr>
<td>Refresh Token</td>
<td>Renseignez le jeton de rafraîchissement requis par Box afin de rafraîchir automatiquement le jeton d’accès. Pour plus d’informations concernant l’obtention du jeton d’accès et du jeton de rafraîchissement, consultez la documentation de Box disponible sur <a href="https://developers.box.com/">https://developers.box.com/</a> (en anglais).</td>
</tr>
<tr>
<td>Use HTTP proxy</td>
<td>Si vous utilisez un proxy, cochez cette case et saisissez les informations d’hôte et de port dans les champs correspondants qui s’affichent.</td>
</tr>
</tbody>
</table>
### Advanced settings

| tStatCatcher Statistics | Cochez cette case afin de collecter les informations de log au niveau du composant. |

### Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant est utilisé en standalone en tant que sous-job afin de créer la connexion à Box utilisée. Dans un Job, il est souvent relié aux autres composants Box à l’aide de liens Trigger tel que le lien OnSubjobOk. |

### Scénario associé

Pour un scénario associé, consultez Scénario : Charger et télécharger des fichiers depuis Box à la page 415.
**tBoxCopy**

Ce composant copie ou déplace un dossier ou un fichier donné au sein de Box.

**Propriétés du tBoxCopy Standard**

Ces propriétés sont utilisées pour configurer le tBoxCopy s'exécutant dans le framework de Jobs Standard.

Le composant tBoxCopy Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Basic settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Connection/Client Key</td>
<td>Saisissez la clé client requise par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur <a href="https://developers.box.com/">https://developers.box.com/</a> puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.</td>
</tr>
<tr>
<td>Connection/Client Secret</td>
<td>Saisissez le Secret du client requis par Box afin d'accéder à l'API de Box. Afin d'obtenir la clé client et le Secret du client, vous devez créer un compte sur <a href="https://developers.box.com/">https://developers.box.com/</a> puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l'application du compte.</td>
</tr>
<tr>
<td>Connection/Access Token</td>
<td>Renseignez le jeton d'accès requis par Box afin d'accéder au compte Box et le faire fonctionner. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez la documentation de Box disponible sur <a href="https://developers.box.com/">https://developers.box.com/</a> (en anglais).</td>
</tr>
<tr>
<td>Connection/Refresh Token</td>
<td>Renseignez le jeton de rafraîchissement requis par Box afin de rafraîchir automatiquement le jeton d'accès. Pour plus d'informations concernant l'obtention du jeton d'accès et du jeton de rafraîchissement, consultez la documentation de Box disponible sur <a href="https://developers.box.com/">https://developers.box.com/</a> (en anglais).</td>
</tr>
<tr>
<td>Connection/Use HTTP proxy</td>
<td>Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port dans les champs correspondants qui s'affichent.</td>
</tr>
<tr>
<td>Move Directory</td>
<td>Cochez cette case afin de déplacer un dossier au sein de Box.</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Copy Directory</td>
<td>Cochez cette case afin de copier un dossier au sein de Box.</td>
</tr>
<tr>
<td>File Name</td>
<td>Saisissez le nom du fichier que vous souhaitez copier ainsi que son chemin dans Box.</td>
</tr>
<tr>
<td>Source Directory</td>
<td>Cette option n'est disponible que si vous avez coché la case <strong>Move Directory</strong> ou la case <strong>Copy Directory</strong>. Saisissez le nom du répertoire source dans Box à copier ou à déplacer.</td>
</tr>
<tr>
<td>Destination Directory</td>
<td>Saisissez le nom du dossier cible dans Box dans lequel vous souhaitez copier ou déplacer le fichier ou le dossier spécifié.</td>
</tr>
<tr>
<td>Rename</td>
<td>Cochez cette case afin de renommer le fichier ou le dossier copié. Lorsque vous copiez un fichier, saisissez le nouveau nom du fichier dans le champ <strong>Destination File Name</strong>. Lorsque vous copiez un dossier, saisissez le nouveau nom du dossier dans le champ <strong>New Directory Name</strong>.</td>
</tr>
<tr>
<td>Remove Source File</td>
<td>Cochez cette case afin de supprimer le fichier source pendant l'action de copie.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <code>line</code> lors du nommage des champs. Notez que le schéma de ce composant est en lecture seule. Il a quatre colonnes nommées <code>destinationFile</code>, <code>destinationFileName</code>, <code>sourceDirectory</code> et <code>destinationDirectory</code>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case afin de collecter les informations de log au niveau du composant. |

**Global Variables**

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **DESTINATION_FILENAME** : nom du fichier cible. Cette variable est une variable **After** et retourne une chaîne de caractères. |
**DESTINATION_FILEPATH** : chemin d’accès au fichier cible. Cette variable est une variable *After* et retourne une chaîne de caractères.

**SOURCE_DIRECTORY** : répertoire source. Cette variable est une variable *After* et retourne une chaîne de caractères.

**DESTINATION_DIRECTORY** : répertoire cible. Cette variable est une variable *After* et retourne une chaîne de caractères.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

---

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en stand-alone dans un sous-job afin de copier ou déplacer des données au sein de Box. |

---

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tBoxDelete**

Ce composant supprime un fichier ou dossier depuis Box.
Ce composant se connecte à un compte Box et supprime un fichier ou un dossier défini.

**Propriétés du tBoxDelete Standard**

Ces propriétés sont utilisées pour configurer le tBoxDelete s’exécutant dans le framework de Jobs Standard.
Le composant tBoxDelete Standard appartient à la famille Cloud.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Connection/Client Key</td>
<td>Saisissez la clé client requise par Box afin d’accéder à l’API de Box. Afin d’obtenir la clé client et le Secret du client, vous devez créer un compte sur <a href="https://developers.box.com/">https://developers.box.com/</a> puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l’application du compte.</td>
</tr>
<tr>
<td>Connection/Access Token</td>
<td>Renseignez le jeton d’accès requis par Box afin d’accéder au compte Box et le faire fonctionner. Pour plus d’informations concernant l’obtention du jeton d’accès et du jeton de rafraîchissement, consultez la documentation de Box disponible sur <a href="https://developers.box.com/">https://developers.box.com/</a> (en anglais).</td>
</tr>
<tr>
<td>Connection/Refresh Token</td>
<td>Renseignez le jeton de rafraîchissement requis par Box afin de rafraîchir automatiquement le jeton d’accès. Pour plus d’informations concernant l’obtention du jeton d’accès et du jeton de rafraîchissement, consultez la documentation de Box disponible sur <a href="https://developers.box.com/">https://developers.box.com/</a> (en anglais).</td>
</tr>
<tr>
<td>Connection/Use HTTP proxy</td>
<td>Si vous utilisez un proxy, cochez cette case et saisissez les informations d’hôte et de port dans les champs correspondants qui s’affichent.</td>
</tr>
</tbody>
</table>
Path
Saisissez le chemin sur Box pointant vers le dossier ou le fichier que vous souhaitez supprimer.

Schema et Edit schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.
Notez que le schéma de ce composant est en lecture seule. Il contient une seule colonne nommée `filepath`.

Advanced settings

Cochez cette case afin de collecter les informations de log au niveau du composant.

Global Variables

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**REMOVED_PATH** : chemin d’accès au dossier ou fichier supprimé sur Box. Cette variable est une variable `Flow` et retourne une chaîne de caractères.

Une variable `Flow` fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

Utilisation

Ce composant est généralement utilisé en standalone dans un sous-job afin de supprimer des données depuis Box.

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tBoxGet

Ce composant télécharge un fichier depuis un compte Box.

Ce composant se connecte à un compte Box donné et télécharge des fichiers vers un répertoire local défini.

**Propriétés du tBoxGet Standard**

Ces propriétés sont utilisées pour configurer le tBoxGet s'exécutant dans le framework de Jobs Standard.

Le composant tBoxGet Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Connection/Client Key</td>
<td>Saisissez la clé client requise par Box afin d’accéder à l’API de Box. Afin d’obtenir la clé client et le Secret du client, vous devez créer un compte sur <a href="https://developers.box.com/">https://developers.box.com/</a> puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l’application du compte.</td>
</tr>
<tr>
<td>Connection/Access Token</td>
<td>Renseignez le jeton d’accès requis par Box afin d’accéder au compte Box et le faire fonctionner. Pour plus d’informations concernant l’obtention du jeton d’accès et du jeton de rafraîchissement, consultez la documentation de Box disponible sur <a href="https://developers.box.com/">https://developers.box.com/</a> (en anglais).</td>
</tr>
<tr>
<td>Connection/Refresh Token</td>
<td>Renseignez le jeton de rafraîchissement requis par Box afin de rafraîchir automatiquement le jeton d’accès. Pour plus d’informations concernant l’obtention du jeton d’accès et du jeton de rafraîchissement, consultez la documentation de Box disponible sur <a href="https://developers.box.com/">https://developers.box.com/</a> (en anglais).</td>
</tr>
<tr>
<td><strong>Connection/Use HTTP proxy</strong></td>
<td>Si vous utilisez un proxy, cochez cette case et saisissez les informations d’hôte et de port dans les champs correspondants qui s’affichent.</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Path</strong></td>
<td>Saisissez le chemin d’accès à Box pointant vers le fichier à télécharger.</td>
</tr>
<tr>
<td><strong>Save as file</strong></td>
<td>Cochez cette case pour afficher le champ <strong>Save To</strong> et par courez votre système, ou saisissez le chemin d’accès au répertoire local dans lequel stocker le fichier téléchargé. Si le fichier existe, il est remplacé.</td>
</tr>
</tbody>
</table>
| **Schema et Edit schema**     | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.  
Notez que le schéma de ce composant est en lecture seule. Il contient deux colonnes nommées **fileName** et **content**.  
Le champ **Schema** n’est disponible que si vous cochez la case **Save as file**. |

**Advanced settings**

| **tStatCatcher Statistics**   | Cochez cette case afin de collecter les informations de log au niveau du composant.                                                                                                               |

**Global Variables**

| **Global Variables** | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
**FILE_NAME** : nom du fichier traité. Cette variable est une variable **Flow** et retourne une chaîne de caractères.  
**INPUT_STREAM** : le contenu du fichier récupéré. Cette variable est une variable **Flow** et retourne une classe InputStream.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**. |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

408
**Utilisation**

| Règle d'utilisation | Ce composant peut être utilisé seul ou avec d'autres composants via un lien **Iterate** ou un lien **Trigger** tel que **OnSubjobOk**. |

**Scénario associé**

Pour un scénario associé, consultez **Scénario : Charger et télécharger des fichiers depuis Box** à la page 415.
**tBoxList**

Ce composant liste des fichiers stockés dans un répertoire défini sur Box.

Ce composant lit un (des) fichier(s) sur Box dans le répertoire défini et liste les métadonnées et les contenus de ce (ces) fichier(s).

**Propriétés du tBoxList Standard**

Ces propriétés sont utilisées pour configurer le tBoxList s’exécutant dans le framework de Jobs Standard.

Le composant tBoxList Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection/Client Key</td>
<td>Saisissez la clé client requise par Box afin d’accéder à l’API de Box. Afin d’obtenir la clé client et le Secret du client, vous devez créer un compte sur <a href="https://developers.box.com/">https://developers.box.com/</a> puis créer une application Box sous le compte Box utilisé. La clé client et le Secret du client peuvent être obtenus depuis les paramètres de l’application du compte.</td>
</tr>
<tr>
<td>Connection/Access Token</td>
<td>Renseignez le jeton d’accès requis par Box afin d’accéder au compte Box et le faire fonctionner. Pour plus d’informations concernant l’obtention du jeton d’accès et du jeton de rafraîchissement, consultez la documentation de Box disponible sur <a href="https://developers.box.com/">https://developers.box.com/</a> (en anglais).</td>
</tr>
<tr>
<td>Connection/Refresh Token</td>
<td>Renseignez le jeton de rafraîchissement requis par Box afin de rafraîchir automatiquement le jeton d’accès. Pour plus d’informations concernant l’obtention du jeton d’accès et du jeton de rafraîchissement, consultez la documentation de Box disponible sur <a href="https://developers.box.com/">https://developers.box.com/</a> (en anglais).</td>
</tr>
<tr>
<td>Connection/Use HTTP proxy</td>
<td>Si vous utilisez un proxy, cochez cette case et saisissez les informations d’hôte et de port dans les champs correspondants qui s’affichent.</td>
</tr>
<tr>
<td>Path</td>
<td>Saisissez le chemin pointant vers le dossier contenant les fichiers à lister ou saisissez le chemin vers le fichier que vous souhaitez lire.</td>
</tr>
</tbody>
</table>
List type
Sélectionnez le type de données que vous souhaitez lister dans le chemin défini : Files, Folders ou Both.

Include subdirectories
Cochez cette case afin de lister des fichiers contenus dans les sous-dossiers en plus de ceux contenus dans le dossier défini dans le champ Path.

Schema et Edit schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Notez que le schéma de ce composant est en lecture seule. Il contient six colonnes nommées name, path, lastModified, size, id et type.

Advanced settings

Global Variables
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
NAME : le nom du fichier distant traité. Cette variable est une variable Flow et retourne une chaîne de caractères.
FILE_PATH : chemin d’accès au dossier ou fichier à traiter sur Box. Cette variable est une variable Flow et retourne une chaîne de caractères.
FILE_DIRECTORY : répertoire du dossier ou fichier à traiter sur Box. Cette variable est une variable Flow et retourne une chaîne de caractères.
LAST_MODIFIED : date et heure de la dernière modification apportée au fichier traité. Cette variable est une variable Flow et retourne un long.
SIZE : le volume du fichier traité. Cette variable est une variable Flow et retourne un long.
ID : ID du dossier ou fichier à traiter sur Box. Cette variable est une variable Flow et retourne une chaîne de caractères.
TYPE : type des objets à traiter sur Box, fichier ou dossier. Cette variable est une variable Flow et retourne une chaîne de caractères.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace.
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé en standalone. |

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tBoxPut**

Ce composant charge des fichiers sur un compte Box.

Ce composant charge des données sur Box à partir d’un fichier local ou à partir d’un flux de données.

**Propriétés du tBoxPut Standard**

Ces propriétés sont utilisées pour configurer le tBoxPut s'exécutant dans le framework de Jobs Standard.

Le composant tBoxPut Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>
Remote Path | Saisissez le chemin pointant vers le fichier dans lequel vous souhaitez écrire des données. Si le fichier n’existe pas, il est créé à la volée.

Replace if Existing | Cochez cette case afin de remplacer le fichier existant par le fichier chargé.

Upload mode | Sélectionnez le mode de chargement utilisé :
- **Upload incoming content as file** : Sélectionnez ce bouton radio afin de lire les données directement depuis le flux d’entrée du composant précédent. Ces données sont ensuite écrites dans le fichier spécifié dans le champ Remote Path.
- **Upload local file** : Sélectionnez ce bouton radio afin de charger un fichier local sur Box. Dans le champ File qui s’affiche, saisissez le chemin ou parcourez votre système jusqu’à ce fichier.
- **Expose as OutputStream** : Cochez cette case afin d’exposer le flux de sortie de ce composant. Ce flux de sortie peut être utilisé par d’autres composants afin d’écrire des données dans le fichier. Par exemple, vous pouvez utiliser la fonctionnalité **Use output stream** du composant tFileOutputDelimited afin d’alimenter le flux de sortie d’un tBoxPut exposé. Pour plus d’informations, consultez tFileOutputDelimited à la page 1169.

Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Notez que le schéma de ce composant est en lecture seule. Il contient une seule colonne nommée content et reçoit des données uniquement depuis la colonne content du schéma d’entrée. Cela signifie que vous devez utiliser une colonne content dans le flux de données d’entrée afin de transporter les données à charger. Ce type de colonne est généralement fourni par le composant tFileInputRaw. Pour plus d’informations, consultez tFileInputRaw à la page 1139.

Le champ Schema n’est disponible que si vous avez sélectionné le mode de chargement **Expose as OutputStream** ou **Upload local file**.

Advanced settings | 

**tStatCatcher Statistics** | Cochez cette case afin de collecter les informations de log au niveau du composant.

Global Variables | 

**Global Variables** | 

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est
une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

---

### Utilisation

**Règle d’utilisation**

Ce composant est utilisé soit en standalone dans un sous-job, afin de charger un fichier local directement sur Box, ou en tant que composant de fin du flux d’un Job afin de charger des données traitées par ce flux.

### Scénario : Charger et télécharger des fichiers depuis Box

Ce scénario décrit un Job de trois composants qui charge un fichier sur Box puis télécharge un fichier depuis Box sur votre système de fichiers local.


### Relier les composants

**Procédure**

1. Dans la perspective **Integration** du Studio, créez un Job vierge depuis le nœud **Job Designs** du Repository.

   Pour plus d’informations sur la création d’un Job, consultez le Guide utilisateur du Studio Talend.
Dans l'espace de modélisation graphique, saisissez le nom du composant à utiliser et sélectionnez ce composant dans la liste qui s'ouvre. Dans ce scénario, les composants sont le **tBoxConnection**, le **tBoxPut** et le **tBoxGet**.

3. Reliez le **tBoxConnection** au **tBoxPut** à l'aide d'un lien **Trigger > OnSubjobOk**.

4. Reliez le **tBoxPut** au **tBoxGet** à l'aide d'un lien **Trigger > OnSubjobOk**.

## Configurez les composants

### Procédure

1. Double-cliquez sur le **tBoxConnection** afin d'ouvrir sa vue **Component**.

2. Renseignez la clé client, le Secret du client, le jeton d'accès et le jeton de rafraîchissement entre guillemets doubles dans les champs correspondants afin d'accéder au compte Box.

3. Double-cliquez sur le **tBoxPut** afin d'ouvrir sa vue **Component**.

4. Cochez la case **Use Existing Connection** afin de réutiliser la connexion créée par le composant **tBoxConnection**.  
Dans le champ **Remote Path**, saisissez le chemin vers lequel vous souhaitez charger le fichier.  
Dans la zone **Upload mode**, sélectionnez **Upload Local File**. Dans le champ **File**, saisissez le chemin ou parcourez votre système vers le fichier que vous souhaitez charger.

5. Double-cliquez sur le **tBoxGet** afin d'ouvrir sa vue **Component**.
6. Cochez la case **Use Existing Connection** afin de réutiliser la connexion créée par le **tBoxConnection**.
   
   Dans le champ **Path**, saisissez le chemin du fichier que vous souhaitez télécharger.
   
   Cochez la case **Save As File**. Dans le champ **Save To**, saisissez le chemin vers lequel vous souhaitez sauvegarder le fichier dans votre système local.

7. Sauvegardez le Job.

**Exécuter le Job**

Exécutez le Job en appuyant sur **F6** ou en cliquant sur le bouton **Run** dans l'onglet **Run**.

Le fichier local, *hello.txt* dans cet exemple, est chargé sur votre Box.

Le fichier *box.txt* est téléchargé depuis Box vers votre système de fichiers local.
**tBufferInput**

Ce composant récupère les données mises en mémoire tampon, via un composant tBufferOutput par exemple, pour les traiter dans un deuxième temps.

Ce composant permet de récupérer les données mises en tampon pour pouvoir les réutiliser dans un deuxième sous-job.

**Propriétés du tBufferInput Standard**

Ces propriétés sont utilisées pour configurer le tBufferInput s'exécutant dans le framework de Jobs Standard.

Le composant tBufferInput Standard appartient à la famille Misc.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• View schema : sélectionnez cette option afin de voir le schéma.  
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].  
Lors de l'utilisation du tBufferInput, c'est l'ordre des colonnes qui est pris en compte, ainsi l'ordre des colonnes est plus important que leur libellé. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Le schéma est créé et conservé pour ce composant uniquement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est le composant de début d’un Job secondaire qui se déclenche automatiquement à la fin du Job principal.</th>
</tr>
</thead>
</table>

Scénario : Récupérer les données mises en mémoire tampon

Ce scénario décrit un Job qui récupère les données mises en mémoire tampon lors de l’exécution d’un premier sous-job et les affiche dans la console standard.

- Cliquez et déposez les composants suivants : tFileInputDelimited et tBufferOutput.
- Sélectionnez le composant tFileInputDelimited et dans sa vue Component, paramétrez les propriétés d’accès au fichier d’entrée.
Dans le champ **File Name**, paramétrez le chemin d’accès au fichier délimité contenant les données à mettre en tampon.

Dans les champs **Row Separator** et **Field Separator**, renseignez respectivement les séparateurs de lignes et de champs.

Dans le champ **Header**, renseignez les informations sur un éventuel en-tête.

Cliquez sur le bouton [...] à côté du champ Edit schema pour décrire la structure du fichier.

**Décrivez le schéma des données à passer au composant tBufferOutput.**

Sélectionnez le composant tBufferOutput et paramétrez ses propriétés dans la vue Component.

Généralement le schéma du composant d’entrée alimente automatiquement le schéma du composant tBufferOutput. Mais vous pouvez aussi configurer une partie du schéma à mettre en tampon si vous le souhaitez.

Déposez les composants tBufferInput et tLogRow en dessous du sous-job que vous venez de créer.

Reliez le composant tFileInputDelimited au tBufferInput via un lien de type Trigger > OnSubjobOk et reliez le composant tBufferInput au tLogRow via un lien de type Row > Main.

Double-cliquez sur le composant tBufferInput pour paramétrer ses propriétés dans la vue Component.

Dans l’onglet Basic settings, cliquez sur le bouton [...] à côté du champ Edit schema pour décrire la structure du fichier.

Utilisez la même description de fichier que celle définie pour le composant tFileInputDelimited et cliquez sur OK.
• Le schéma du composant `tBufferInput` alimente automatiquement le schéma du composant `tLogRow`. Sinon double-cliquez sur le `tLogRow` pour afficher ses propriétés dans la vue `Component` et cliquez sur le bouton `Sync columns`.

• Enregistrez votre Job et appuyez sur `F6` pour l’exécuter.

Starting job tBufferInput_scenario at 14:13 12/06/2009.

La console affiche les données qui ont été récupérés de la mémoire tampon.
tBufferOutput

Ce composant met en tampon des données afin de pouvoir y accéder plus tard via un service Web par exemple.

Le tBufferOutput a été conçu pour être exporté en tant que service Web afin d’accéder aux données directement à partir du serveur d’application Web. Pour plus d’informations, consultez le Guide utilisateur du Studio Talend.

Propriétés du tBufferOutput Standard

Ces propriétés sont utilisées pour configurer le tBufferOutput s’exécutant dans le framework de Jobs Standard.

Le composant tBufferOutput Standard appartient à la famille Misc.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
| | • View schema : sélectionnez cette option afin de voir le schéma.
| | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
| | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
| | Lors de l’utilisation du tBufferOutput, c’est l’ordre des colonnes qui est pris en compte, ainsi l’ordre des colonnes est plus important que leur libellé.
**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métdonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant n’est pas un composant de début (font vert) et il requiert un composant de sortie. |

**Scénario 1 : Mettre des données en mémoire tampon**

Ce scénario décrit un Job volontairement basique qui met en tampon les données d’un Job enfant pendant qu’un Job parent affiche les données mise en tampon dans la console standard. Généralement, un Job utilisant un composant tBufferOutput inclurait une étape d’export en tant que service Web, afin que les données soient accessibles directement via le serveur d’application Web.
Créez deux Jobs : le premier Job (*BufferFatherJob*) exécute le deuxième et affiche le contenu dans la console Run. Le deuxième Job (*BufferChildJob*) stocke les données définies dans une mémoire tampon.

Dans le premier Job, cliquez et déposez les composants suivants : *tRunJob* et *tLogRow*.

Dans le deuxième Job, cliquez et déposez les composants suivants : *tFileInputDelimited* et *tBufferOutput*.

Commencez par paramétrer les propriétés du deuxième Job.

Sélectionnez le composant *tFileInputDelimited* et dans l'onglet *Basic settings* de la vue *Component*, paramétrez les propriétés d'accès au fichier d'entrée.

Dans le champ *File Name*, paramétrez le chemin d'accès au fichier délimité contenant les données à mettre en tampon.

Dans les champs *Row Separator* et *Field Separator*, renseignez respectivement les séparateurs de lignes et de champs.

Dans le champ *Header*, renseignez les informations sur un éventuel en-tête.

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Type</th>
<th>Nullable</th>
<th>Date Part...</th>
<th>Length</th>
<th>Prec...</th>
<th>Dec...</th>
<th>Comp...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Postal</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>State</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capital</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MostPopulousCity</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• Décrivez le **Schema** des données à passer au composant **tBufferOutput**.

• Sélectionnez le composant **tBufferOutput** et paramétrez ses propriétés dans l'onglet **Basic settings** de la vue **Component**.

• Généralement le schéma du composant d’entrée alimente automatiquement le schéma du composant **tBufferOutput**. Mais vous pouvez aussi configurer une partie du schéma à mettre en tampon si vous le souhaitez.

• Maintenant le premier Job (**BufferFatherJob**), paramétrez les propriétés du composant **tRunJob**.

  ![Schema Type](Built-In) ![Edit schema](Sync columns)

• Cliquez sur **Edit schema** pour modifier le schéma si nécessaire et sélectionner la colonne à afficher. Le schéma peut être identique ou différent du schéma mis en tampon.

• Dans le tableau **Context Param**, vous pouvez aussi définir les paramètres de contexte à utiliser pour l’exécution de ce Job. Pour cet exemple, gardez le contexte par défaut.

Appuyez sur **F6** pour exécuter les Jobs parent. Le composant **tRunJob** se charge d’exécuter le Job enfant et affiche les données dans la console standard:

Starting job BufferFatherJob at 10:41 05/02/2008.
AL|Alabama|Montgomery|Birmingham
AK|Alaska|Juneau|Anchorage
AZ|Arizona|Phoenix|Phoenix
AR|Arkansas|Little Rock|Little Rock
CA|California|Sacramento|Los Angeles
CO|Colorado|Denver|Denver
CT|Connecticut|Hartford|Bridgeport
DE|Delaware|Dover|Wilmington
FL|Florida|Tallahassee|Jacksonville
GA|Georgia|Atlanta|Atlanta
HI|Hawaii|Honolulu|Honolulu
ID|Idaho|Boise|Boise
IL|Illinois|Springfield|Chicago
IN|Indiana|Indianapolis|Indianapolis
IA|Iowa|Des Moines|Des Moines
KS|Kansas|Topeka|Wichita
KY|Kentucky|Frankfort|Louisville
MI|Michigan|Detroit|Grand Rapids
Mettre en mémoire tampon des données à utiliser en tant que système source

Ce scénario décrit un Job mettant en mémoire tampon des données à utiliser en tant que système source par MDM.

Un processus MDM invoque ce Job pour récupérer les données en recherchant les éléments définis (valeurs des régions des agents) dans les données mises en mémoire tampon. Le processus peut afficher les données récupérées dans Talend MDM Web User Interface, sans les sauvegarder réellement dans le hub MDM.

Créer un Job mettant des données en mémoire tampon

**Procédure**
1. Créez un Job nommé `DetermineRegion`.
2. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : un `tJava`, un `tFixedFlowInput` et un `tBufferOutput`.
3. Reliez le `tJava` au `tFixedFlowInput` à l'aide d’un lien `Trigger > On Component Ok`.
4. Reliez le `tFixedFlowInput` au `tBufferOutput` à l’aide d’un lien `Row > Main`.

Configurer le Job pour mettre des données en mémoire tampon

**Procédure**
1. Dans la vue `Contexts`, ajoutez une nouvelle variable de contexte. Dans le champ `Name`, saisissez `xmlInput` et, dans la colonne `Type`, sélectionnez `String`.

   Dans cet exemple, la variable de contexte `xmlInput` du Job sera spécifiée dans le processus MDM souhaitant invoquer ce Job.


   Si vous ne trouvez pas la vue `Contexts`, allez dans le menu `Window > Show view > Talend` et sélectionnez `Contexts`.

   Pour plus d’informations concernant la définition des variables de contexte, consultez le Guide utilisateur du Studio Talend.

   Vous pouvez obtenir plus d’informations concernant la définition de variables de contexte sur Talend Help Center (https://help.talend.com).

2. Double-cliquez sur le composant `tJava` pour ouvrir sa vue `Component` et, dans la zone `Code`, saisissez du code, selon vos besoins.
Dans cet exemple, saisissez `System.out.println("########################################"+context.xmlInput);`.

4. Cliquez sur le bouton ` [...] ` à côté du champ `Edit schema` pour ouvrir la fenêtre du schéma et le définir pour les données à utiliser par le système source.
   Dans cet exemple, ajoutez une colonne `col0` de type `String`.
5. Une fois le schéma défini, cliquez sur `Yes` dans la boîte de dialogue `Propagate` afin de propager les modifications du schéma au composant suivant `tBufferOutput`.
6. Dans le champ `Number of rows`, saisissez `1`.
7. Dans la zone `Mode`, sélectionnez `Use Single Table` et saisissez `"Paris"` dans la colonne `Value` correspondant à la colonne `col0` définie.
   Dans cet exemple, la valeur de `col0` fournit les informations relatives aux régions des agents à récupérer par MDM.
8. Double-cliquez sur le composant `tBufferOutput` pour ouvrir sa vue `Component` et assurez-vous que son schéma et celui du composant précédent `tFixedFlowInput` sont synchronisés.
9. Exécutez le Job et vérifiez que l’exécution est réussie.

### Scénario 2 : Mettre les données de sortie en mémoire tampon du serveur d’application Web

Ce scénario décrit un Job appelant un service Web et stockant les données de sortie directement sur la mémoire tampon du serveur d’application Web. Ce scénario crée d’abord un Job utilisant des variables de contexte qui pourra ensuite être utilisé en tant que service Web, puis exporte de Job en tant que service Web.

#### Créer un Job

**Procédure**

2. Reliez-les via une connexion de type `Row Main`.

**Résultats**

![Diagramme d'implémentation de Job](image)

**Créer une variable de contexte**

**Pourquoi et quand exécuter cette tâche**

Dans ce scénario, vous allez définir deux variables de contexte : `nb_lines` et `lastname`. La première variable configure le nombre de lignes générées par le composant `tFixedFlowInput` et la seconde
configure le nom de famille à afficher dans la liste de sortie. Pour plus d’informations concernant la création et l’utilisation de variables de contexte, consultez le Guide utilisateur de Studio Talend.

Pour définir les deux variables de contexte :

**Procédure**

1. Dans l’onglet **Contexts** de votre Job, cliquez sur le bouton [+] au bas de la vue, afin d’ajouter deux variables, respectivement **nb_lines**, de type **Integer** et **lastname** de type **String**.
2. Dans le champ **Value** des variables, configurez le nom de famille à afficher et le nombre de lignes à générer, respectivement **Ford** et **3** dans cet exemple.

![Configuration des variables de contexte](image)

**Configurer les données d’entrée**

**Procédure**

1. Dans l’espace de travail, sélectionnez le **tFixedFlowInput**.
2. Cliquez sur la vue **Component** pour paramétrer ses propriétés de base dans l’onglet **Basic settings**.
3. Cliquez sur le bouton [...] à côté du champ **Edit Schema** pour décrire la structure des données que vous souhaitez à partir des variables internes. Dans ce scénario, le schéma est constitué de trois colonnes : **now**, de type **Date**, **firstname** et **lastname**, toutes de type **String**.

![Schéma des données d'entrée](image)

4. Cliquez sur **OK** pour fermer la boîte de dialogue et acceptez la propagation des modifications lorsque vous y êtes invité. Les trois colonnes définies apparaissent dans le champ **Values** de l’onglet **Basic settings** du **tFixedFlowInput**.
5. Cliquez dans la cellule **Value** de chaque ligne des deux premier paramètres définis et appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales.

6. A partir de la liste des variables globales, sélectionnez `TalendDate.getCurrentDate()` pour la colonne `now` et `talendDatagenerator.getFirstName()` pour la colonne `firstname`.

7. Cliquez dans la cellule **Value** de la colonne `lastname` et appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales.

8. A partir de la liste des variables globales, sélectionnez `context.lastname`, la variable de contexte que vous avez créé pour la colonne `lastname`.

### Résultats

![Diagramme de la tâche](image)

### Construire un Job en tant que service Web

**Pourquoi et quand exécuter cette tâche**

Avant de construire votre Job en tant que service Web, consultez le Guide utilisateur du Studio Talend.

**Procédure**

1. Dans la vue **Repository**, cliquez-droit sur le Job que vous venez de créer et sélectionnez **Build Job** dans le menu. La boîte de dialogue `[Build Job]` s'ouvre.
2. Cliquez sur le bouton **Browse...** et sélectionnez le répertoire dans lequel enregistrer votre Job.

3. Dans la zone **Build type**, sélectionnez le type de construction (build) que vous souhaitez utiliser dans votre application Web Tomcat (WAR pour cet exemple) et cliquez sur **Finish**. La boîte de dialogue [Build Job] se referme.

4. Copiez le fichier War et collez-le dans le répertoire de votre Tomcat.

**Scénario 3 : Appeler un Job contenant des variables de contexte à partir de votre navigateur Web**

Ce scénario décrit la procédure à utiliser pour appeler le Job créé dans **Scénario 2 : Mettre les données de sortie en mémoire tampon du serveur d’application Web** à la page 427 à partir de votre navigateur Web en modifiant/sans modifier les valeurs des variables de contexte.

Cliquez sur **Enter** pour exécuter votre Job à partir du navigateur.

- `<soapenv:Envelope>
  - `<soapenv:Body`
    - `<runJobReturn xsi:type="ns1:runJobReturn">
      - `<ns1:item xsi:type="ns1:ArrayOf_xsd_string">
        - `<ns1:item xsi:type="xsd:string">31-07-2008</ns1:item>
        - `<ns1:item xsi:type="xsd:string">William</ns1:item>
        - `<ns1:item xsi:type="xsd:string">Ford</ns1:item>
      </ns1:item>
      - `<ns1:item xsi:type="ns1:ArrayOf_xsd_string">
        - `<ns1:item xsi:type="xsd:string">31-07-2008</ns1:item>
        - `<ns1:item xsi:type="xsd:string">Milard</ns1:item>
        - `<ns1:item xsi:type="xsd:string">Ford</ns1:item>
      </ns1:item>
      - `<ns1:item xsi:type="ns1:ArrayOf_xsd_string">
        - `<ns1:item xsi:type="xsd:string">31-07-2008</ns1:item>
        - `<ns1:item xsi:type="xsd:string">James</ns1:item>
        - `<ns1:item xsi:type="xsd:string">Ford</ns1:item>
      </ns1:item>
    </runJobReturn>
  </soapenv:Body>
</soapenv:Envelope>

Le Job utilise les valeurs par défaut des variables de contexte : *nb_lines* et *lastname*, et donc génère trois lignes contenant la date en cours, un prénom et le nom Ford.

Vous pouvez modifier les valeurs des variables de contexte directement à partir de votre navigateur. Pour appeler le Job à partir de votre navigateur et modifier les valeurs des deux variables de contexte, saisissez l’URL suivante :

```
```

%20 correspond à un espace blanc lorsqu’il est utilisé dans une URL. Dans le premier argument "arg1", définissez la valeur "MASSY" pour la variable de contexte *lastname*. Ainsi le nom qui apparaîtra sera "MASSY". Dans le deuxième argument "arg2", définissez la valeur "2" pour la variable de contexte *nb_lines* pour ne générer que deux lignes de données.

Cliquez sur **Enter** pour exécuter le Job à partir du navigateur.
Le Job génère deux lignes avec MASSY comme nom.

Scénario 4 : Appeler un Job exporté en tant que service Web dans un autre Job

Ce scénario décrit un Job appelant un autre Job exporté en tant que service Web en utilisant un composant tWebServiceInput. Ce scénario appellera le Job créé dans Scénario 2 : Mettre les données de sortie en mémoire tampon du serveur d’application Web à la page 427.

- Cliquez-déposez les composants suivants dans le Job Designer : tWebServiceInput et tLogRow.
- Reliez-les via une connexion de type Row Main.

Dans l’espace de travail, sélectionnez le composant tWebServiceInput.
- Cliquez sur la vue Component pour définir ses propriétés de base dans l’onglet Basic settings.
- Dans la liste Schema, sélectionnez Built-In et cliquez sur le bouton [...] à côté du champ Edit Schema pour décrire la structure des données que vous souhaitez appeler du Job exporté. Dans ce scénario, le schéma est constitué de trois colonnes, now, firstname, et lastname.
• Cliquez sur le bouton [+] pour ajouter les trois lignes de paramètres et définir vos variables. Cliquez sur OK pour fermer la boîte de dialogue.

• Dans le champ WSDL de l’onglet Basic settings de la vue Component du tWebServiceInput, saisissez l’URL : http://localhost:8080/export_job/services/export_job3?WSDL dans laquelle "export_job" correspond au nom du répertoire de votre application Web dans lequel le Job à appeler est stocké et "export_job3" correspond au nom du Job.

• Dans le champ Method name, saisissez runJob.

• Dans la zone Parameters, cliquez sur le bouton [+] pour ajouter deux lignes de paramètres et définir vos variables de contexte.

• Cliquez dans la première cellule Value pour configurer le paramètre définissant le nombre de lignes à générer : --context_param nb_line=3.

• Cliquez dans la deuxième cellule Value pour configurer le paramètre définissant le nom à afficher : --context_param lastname=Ford.

• Sélectionnez le tLogRow et cliquez sur la vue Component pour afficher les propriétés du composant.
Dans la vue **Basic settings** du **tLogRow**, sélectionnez l’option **Table** pour afficher les données de sortie sous forme de tableau. Pour plus d’informations, consultez Propriétés du **tLogRow Standard** à la page 2105.

Enregistrez votre Job et appuyez sur **F6** pour l’exécuter.

```
Starting job Call_Webservice_In_Job at 14:12 01/08/2008.

<table>
<thead>
<tr>
<th>tLogRow_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>now</td>
</tr>
<tr>
<td>01-08-2008</td>
</tr>
<tr>
<td>01-08-2008</td>
</tr>
<tr>
<td>01-08-2008</td>
</tr>
</tbody>
</table>

Job Call_Webservice_In_Job ended at 14:12 01/08/2008 [exit code=0]
```

Le Job génère trois colonnes contenant la date en cours, des prénoms et des noms, et les affiche dans la console sous forme de tableau.
**tCassandraBulkExec**

Ce composant permet un gain de performance pendant les opérations d'Insert dans une famille de colonnes Cassandra.

Les tCassandraOutputBulk et tCassandraBulkExec sont généralement utilisés ensemble en tant que parties d'un processus de deux étapes. Dans la première étape, une table SSTable est générée. Dans la seconde étape, cette table SSTable est écrite dans Cassandra. Ces deux étapes sont fusionnées dans le composant tCassandraOutputBulkExec, détaillé dans une section séparée. L’avantage d’utiliser deux composant séparés réside dans le fait que les données peuvent être transformées avant d’être chargées dans Cassandra.

Le tCassandraBulkExec écrit des données d’une table SSTable dans Cassandra.

**Propriétés du tCassandraBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tCassandraBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tCassandraBulkExec Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| DB Version | Sélectionnez la version de Cassandra que vous utilisez.  
**Avertissement** :  
- Pour Cassandra 1.1.2, seules les visites en local sont supportées sous Linux.  
- Pour Cassandra 1.2.2, assurez-vous que le serveur et le client fonctionnent sous le même système d’exploitation.  
- Cassandra 2.0.0 fonctionne uniquement avec la JVM 1.7. |
| Host | Saisissez le nom de l’hôte ou l’adresse IP du serveur Cassandra. |
| Port | Saisissez le numéro du port d’écoute du serveur Cassandra. |
| Required authentication | Cochez cette case afin de fournir les informations d’authentification pour Cassandra.  
**Avertissement** :  
Cette case n’apparaît pas si vous sélectionnez Cassandra 1.1.2 dans la liste DB Version. |
| Username | Saisissez dans ce champ le nom d’utilisateur pour l’authentification à Cassandra. |
**Password**

Saisissez dans ce champ le mot de passe pour l’authentification à Cassandra.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Use configuration file**

Cochez cette case et, dans le champ qui s’affiche, saisissez le chemin d’accès ou parcourez votre système jusqu’au fichier principal de configuration pour Cassandra, `cassandra.yaml`.

Ainsi, ce composant peut importer et utiliser directement la configuration du fichier `cassandra.yaml`, qui peut contenir des propriétés avancées de Cassandra, comme les propriétés relatives au chiffrement SSL.

Lorsque vous devez exécuter votre Job dans différents environnements Cassandra, cette fonctionnalité permet à votre Job de passer facilement d’une configuration à une autre.

Pour plus d’informations concernant ce fichier `cassandra.yaml`, consultez [Cassandra configuration](en anglais).

**Keyspace**

Saisissez le nom du Keyspace dans lequel écrire la table SSTable.

**Column family**

Saisissez le nom de la famille de colonnes dans laquelle écrire la table SSTable.

**SSTable directory**

Spécifiez le répertoire local de la table SSTable à charger dans Cassandra. Notez que le chemin d’accès complet à la table SSTable comprend le répertoire local à la suite du Keyspace spécifié et du nom de la famille de colonnes.

Par exemple, si vous configurez le répertoire local à `/home/talend/sstable` et que vous spécifiez `testk` comme nom du Keyspace et `testc` comme nom de la famille de colonnes, le chemin d’accès complet à la table SSTable est `/home/talend/sstable/testk/testc/`.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

`ERROR_MESSAGE` : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant à cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est utilisé en standalone.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>L’exécution de ce composant termine le Job.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tCassandraClose

Ce composant est utilisé pour arrêter une connexion active à un serveur Cassandra afin de libérer des ressources occupées.

**Propriétés du tCassandraClose Standard**

Ces propriétés sont utilisées pour configurer le tCassandraClose s’exécutant dans le framework de Jobs Standard.

Le composant tCassandraClose Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Component List | Sélectionnez une connexion active Cassandra à fermer. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau des composants. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Cassandra, notamment le tCassandraConnection. |
Scénario associé

Pour un scénario utilisant le tCassandraClose, consultez Scénario : Gestion des données avec Cassandra à la page 447.
tCassandraConnection

Ce composant réutilise la connexion à un serveur Cassandra créée.
Le composant tCassandraConnection ouvre une connexion à un serveur Cassandra.

Propriétés du tCassandraConnection Standard

Ces propriétés sont utilisées pour configurer le tCassandraConnection s’exécutant dans le framework de Jobs Standard.
Le composant tCassandraConnection Standard appartient aux familles Big Data et Databases.
Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Property type          | Peut être Built-In ou Repository.  
|                       | Built-In : propriétés utilisées ponctuellement.  
|                       | Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.  
| Use existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
| DB Version            | Sélectionnez la version de Cassandra que vous utilisez.  
| API type              | Cette liste déroulante s’affiche uniquement lorsque vous avez sélectionné la version 2.0 de Cassandra dans la liste DB version. Dans cette liste API type, vous pouvez sélectionner Datastax pour utiliser CQL 3 (Cassandra Query Language) avec Cassandra ou sélectionnez Hector pour utiliser CQL 2.  
|                       | Notez que l’API Hector est dépréciée depuis la version 2.0 de Cassandra, mais est toujours disponible pour une utilisation dans le Studio, à des fins de flexibilité de version du langage de requête à utiliser avec Cassandra 2.0.0.  
| Server                | Saisissez le nom de l’hôte ou l’adresse IP du serveur Cassandra.  
| Port                  | Saisissez le numéro du port d’écoute du serveur Cassandra.  
| Required authentication | Cochez cette case pour activer l’authentification à la base de données.  
| Username              | Saisissez dans ce champ l’identifiant d’authentification à Cassandra.  

Password

Saisissez dans ce champ le mot de passe d’authentification à Cassandra.

Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

Use SSL connection

Cochez cette case pour activer la connexion chiffrée SSL ou TLS.

Utilisez le composant tSetKeystore dans le même Job afin de spécifier les informations de chiffrement.

Pour plus d’informations concernant le tSetKeystore, consultez tSetKeystore à la page 3745.

Notez que la connexion SSL est disponible uniquement à partir de la version 2.4 de MongoDB.

Advanced settings

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

Global Variables

Global Variables

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Ce composant est généralement utilisé avec d’autres composants Cassandra, notamment le tCassandraClose.

Scénario associé

Pour un scénario dans lequel le tCassandraConnection est utilisé, consultez Scénario : Gestion des données avec Cassandra à la page 447.
tCassandralInput

Ce composant extrait les données souhaitées d’une famille de colonnes ou d’une super famille de colonnes d’un Keyspace Cassandra, afin d’appliquer les modifications aux données.

Le composant tCassandralInput vous permet de lire des données d’un espace de clé (keyspace) Cassandra et d’envoyer des données dans un flux Talend.

Tableaux des correspondances entre le type Cassandra et le type de données Talend

Le premier tableau ci-dessous présente les relations de mapping entre le type Cassandra avec sa nouvelle API, Datastax et le type de données Talend.

<table>
<thead>
<tr>
<th>Type Cassandra</th>
<th>Type de données Talend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ascii</td>
<td>String; Character</td>
</tr>
<tr>
<td>Bigint</td>
<td>Long</td>
</tr>
<tr>
<td>Blob</td>
<td>Byte[]</td>
</tr>
<tr>
<td>Boolean</td>
<td>Boolean</td>
</tr>
<tr>
<td>Counter</td>
<td>Long</td>
</tr>
<tr>
<td>Inet</td>
<td>Object</td>
</tr>
<tr>
<td>Int</td>
<td>Integer; Short; Byte</td>
</tr>
<tr>
<td>List</td>
<td>List</td>
</tr>
<tr>
<td>Map</td>
<td>Object</td>
</tr>
<tr>
<td>Set</td>
<td>Object</td>
</tr>
<tr>
<td>Text</td>
<td>String; Character</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Date</td>
</tr>
<tr>
<td>UUID</td>
<td>String</td>
</tr>
<tr>
<td>TimeUUID</td>
<td>String</td>
</tr>
<tr>
<td>VarChar</td>
<td>String; Character</td>
</tr>
<tr>
<td>VarInt</td>
<td>Object</td>
</tr>
<tr>
<td>Boolean</td>
<td>Boolean</td>
</tr>
<tr>
<td>Float</td>
<td>Float</td>
</tr>
</tbody>
</table>
### Propriétés du tCassandraInput Standard

Ces propriétés sont utilisées pour configurer le tCassandraInput s’exécutant dans le framework de Jobs Standard.

Le composant tCassandraInput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-In : propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong> : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
<td></td>
</tr>
<tr>
<td><strong>Use existing connection</strong> : Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <em>Component List</em> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
<td></td>
</tr>
<tr>
<td><strong>DB Version</strong> : Sélectionnez la version de Cassandra que vous utilisez.</td>
<td></td>
</tr>
<tr>
<td><strong>API type</strong> : Cette liste déroulante s’affiche uniquement lorsque vous avez sélectionné la version 2.0 de Cassandra dans la liste <em>DB version</em>. Dans cette liste <em>API type</em>, vous pouvez sélectionner <em>Datastax</em> pour utiliser CQL 3 (Cassandra Query Language) avec Cassandra ou sélectionnez <em>Hector</em> pour utiliser CQL 2. Notez que l’API <em>Hector</em> est dépréciée depuis la version 2.0 de Cassandra, mais est toujours disponible pour une utilisation dans le Studio, à des fins de flexibilité de version du langage de requête à utiliser avec Cassandra 2.0.0. Avec l’évolution des commandes CQL, les paramètres à configurer dans la vue <em>Basic settings</em> varient.</td>
<td></td>
</tr>
<tr>
<td><strong>Host</strong> : Saisissez le nom de l’hôte ou l’adresse IP du serveur Cassandra.</td>
<td></td>
</tr>
<tr>
<td><strong>Port</strong> : Saisissez le numéro du port d’écoute du serveur Cassandra.</td>
<td></td>
</tr>
<tr>
<td><strong>Required authentication</strong> : Cochez cette case afin de saisir vos identifiants de connexion à Cassandra. Cette case apparaît si vous ne cochez pas la case <em>Use existing connection</em>.</td>
<td></td>
</tr>
<tr>
<td><strong>Username</strong> : Saisissez dans ce champ l’identifiant d’authentification à Cassandra.</td>
<td></td>
</tr>
<tr>
<td><strong>Password</strong> : Saisissez dans ce champ le mot de passe d’authentification à Cassandra. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <em>Password</em>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
<td></td>
</tr>
<tr>
<td><strong>Keyspace</strong> : Saisissez le nom du Keyspace duquel lire les données.</td>
<td></td>
</tr>
<tr>
<td><strong>Column family</strong> : Saisissez le nom de famille de la colonne de laquelle lire les données.</td>
<td></td>
</tr>
<tr>
<td><strong>Keyspace</strong> : Saisissez le nom de l’espace de clé duquel vous souhaitez lire les données.</td>
<td></td>
</tr>
<tr>
<td><strong>Column family</strong> : Saisissez le nom de la famille de colonnes de laquelle vous souhaitez lire les données.</td>
<td></td>
</tr>
</tbody>
</table>
### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Query

Saisissez les instructions de requête à utiliser pour lire les données dans la base de données Cassandra.

Par défaut, la requête n’est pas sensible à la casse. Cela signifie, que, lors de l’exécution, le nom des colonnes saisis dans la requête sont toujours pris en minuscules. Si vous voulez rendre la requête sensible à la casse, mettez les noms de colonnes entre guillemets doubles.

Le bouton `[…]` à côté de ce champ vous permet de générer le code d’exemple affichant quelles sont les variables prédéfinies pour les données à lire et comment ces variables peuvent être utilisées.

Cette fonctionnalité est disponible uniquement à partir de la version 2.0 de l’API Datastax de Cassandra.

### Column family type

- **Standard** : Famille de colonnes de type standard.
- **Super** : Famille de colonnes de type super.

### Include key in output columns

Cochez cette case pour inclure la clé de la famille de colonnes dans les colonnes de sortie.

- **Key column** : sélectionnez le colonne de clé dans la liste.

### Row key type

 Sélectionnez dans la liste le type de données Talend approprié pour la clé de ligne.

### Row key Cassandra type

 Sélectionnez dans la liste le type Cassandra pour la clé de ligne.

**Avertissement :**

La valeur de l’option **Default** varie selon le type de clé de ligne sélectionné. Par exemple,
Si vous sélectionnez *String* dans la liste **Row key type**, la valeur de l’option *Default* est *UTF8*.

Pour plus d’informations concernant les correspondances de mappage entre le type Cassandra et le type de données *Talend*, consultez **Tableaux des correspondances entre le type Cassandra et le type de données Talend** à la page 442.

| **Include super key output columns** | Cochez cette case pour inclure la super clé de la famille de colonnes dans les colonnes de sortie.

- **Super key column** : sélectionnez dans la liste la colonne ayant la super clé souhaitée.

Cette case apparaît si vous sélectionnez *Super* dans la liste **Column family type**.

| **Super column type** | Sélectionnez dans la liste le type de la super colonne.

| **Super column Cassandra type** | Sélectionnez le type Cassandra correspondant à la super colonne, dans la liste.

Pour plus d’informations concernant les correspondances de mappage entre le type Cassandra et le type de données Talend, consultez **Tableaux des correspondances entre le type Cassandra et le type de données Talend** à la page 442.

| **Specify row keys** | Cochez cette case afin de spécifier les clés des lignes de la famille de colonnes.

| **Row Keys** | Saisissez les clés de lignes spécifiques dans la famille de colonnes, au format correct selon le type de clé de ligne.

Ce champ apparaît si vous cochez la case **Specify row keys**.

| **Key start** | Saisissez la clé de la ligne de début du type de données correct.

| **Key end** | Saisissez la clé de la ligne de fin du type de données correct.

| **Key limit** | Saisissez le nombre de lignes à lire entre la ligne de début et la ligne de fin.

| **Specify columns** | Cochez cette case afin de spécifier directement le nom des colonnes de la famille de colonnes.

| **Columns** | Spécifiez directement le nom des colonnes de la famille de colonnes, au format correct selon le type de colonne.

Ce champ apparaît si vous cochez la case **Specify columns**.

| **Columns range start** | Saisissez le nom de la colonne de début du type de données correct.

| **Columns range end** | Saisissez le nom de la colonne de fin du type de données correct. |
Columns range limit

| Columns range limit | Saisissez le nombre de colonnes à lire entre la colonne de début et la colonne de fin. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est un composant d’entrée et nécessite un lien de sortie. |

Scénario : Gestion des données avec Cassandra

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario décrit un Job simple qui lit des données relatives à des employés dans un fichier CSV, écrit les données dans un keyspace Cassandra, extrait les informations personnelles de certains employés, puis affiche les informations dans la console.
Ce scénario nécessite six composants :

- un `tCassandraConnection` pour ouvrir une connexion au serveur Cassandra.
- un `tFileInputDelimited` pour lire le fichier d'entrée, définir la structure des données et les envoie au composant suivant.
- un `tCassandraOutput` pour écrire les données reçues du composant précédent dans un espace de clé (keyspace) Cassandra.
- un `tCassandraInput` pour lire les données du keyspace Cassandra.
- un `tLogRow` pour afficher dans la console les données reçues du composant précédent.
- et un `tCassandraClose` pour fermer la connexion au serveur Cassandra.

**Déposer et relier les composants**

**Procédure**

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : un `tCassandraConnection`, un `tFileInputDelimited`, un `tCassandraOutput`, un `tCassandraInput`, un `tLogRow` et un `tCassandraClose`.
2. Connectez le `tFileInputDelimited` au `tCassandraOutput` à l'aide d'un lien `Row > Main`.
3. Répétez l'opération pour relier le `tCassandraInput` au `tLogRow`.
4. Connectez le `tCassandraConnection` au `tFileInputDelimited` à l'aide d'un lien `Trigger > OnSubjobOk`. 
5. Répétez l’opération pour relier le tFileInputDelimited au tCassandraInput et le tCassandraInput au tCassandraClose.
6. Renommez les composants afin de mieux identifier leur rôle.

**Configurer les composants**

**Ouvrir une connexion Cassandra**

**Procédure**

1. Double-cliquez sur le **tCassandraConnection** pour ouvrir sa vue **Basic settings**.

2. Sélectionnez la version de Cassandra que vous utilisez, dans la liste **DB Version**. Dans cet exemple, sélectionnez **Cassandra 1.1.2**.
3. Dans le champ **Server**, saisissez le nom de l’hôte ou l’adresse IP du serveur Cassandra. Dans cet exemple, saisissez **localhost**.
4. Dans le champ **Port**, saisissez le numéro du port d’écoute du serveur Cassandra.
5. Si nécessaire, saisissez vos informations d’authentification à Cassandra : votre identifiant dans le champ **Username** et votre mot de passe dans le champ **Password**.

**Lire les données d’entrée**

**Procédure**

1. Double-cliquez sur le composant **tFileInputDelimited** pour ouvrir sa vue **Component**.

2. Cliquez sur le bouton [...] à côté du champ **File Name/Stream** afin de parcourir votre système jusqu’au fichier duquel vous souhaitez lire les données. Dans ce scénario, le répertoire est D:/Input/Employees.csv. Le fichier CSV contient quatre colonnes : id, age, name et ManagerID.

   id;age;name;ManagerID 1;20;Alex;1  2;40;Peter;1  3;25;Mark;1  4;26;Michael;1  5;30;Christophe;2  6;26;Stephane;3  7;37;Cedric;3  8;52;Bill;4  9;43;Jack;2  10;28;Andrews;4
3. Dans le champ Header, saisissez 1 pour ignorer la première ligne du fichier CSV (car c’est une ligne d’en-tête).


Écrire des données dans un keyspace Cassandra

Procédure

1. Double-cliquez sur le tCassandraOutput pour ouvrir sa vue Basic settings.

2. Saisissez les informations requises pour la connexion ou cochez la case Use existing connection afin d’utiliser la connexion précédemment configurée. Dans ce scénario, la case Use existing connection est cochée.

3. Dans la zone Keyspace configuration, saisissez le nom du keyspace : Employee, dans cet exemple et sélectionnez Drop keyspace if exists and create dans la liste Action on keyspace.

4. Dans la zone Column family configuration, saisissez le nom de la famille de colonnes : Employee_Info, dans cet exemple et sélectionnez Drop column family if exists and create dans la liste Action on column family.

   La case Define column family structure s’affiche. Dans cet exemple, décochez la case.

5. Dans la liste Action on data, sélectionnez l’action que vous souhaitez effectuer, Upsert dans cet exemple.

6. Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent.

7. Sélectionnez la colonne de clé dans la liste Key column. Dans cet exemple, sélectionnez id.
Si nécessaire, cochez la case **Include key in columns**.

**Lire des données d’un keyspace Cassandra**

**Procédure**

1. Double-cliquez sur le composant *tCassandraInput* pour ouvrir sa vue **Component**.

2. Saisissez les informations requises pour la connexion ou cochez la case **Use existing connection** afin d’utiliser la connexion précédemment configurée. Dans ce scénario, la case **Use existing connection** est cochée.


4. Dans la zone **Column family configuration**, saisissez le nom de la famille de colonnes : *Employee_Info*, dans cet exemple.

5. Sélectionnez **Edit schema** pour définir la structure des données à lire depuis le keyspace Cassandra. Dans cet exemple, trois colonnes sont définies *id*, *name* et *age*.
6. Si nécessaire, cochez la case Include key in output columns, puis sélectionnez, dans la liste Key column, la colonne clé de la famille de colonnes que vous souhaitez inclure.

7. Dans la liste Row key type, sélectionnez Integer car la colonne id est de type Integer dans cet exemple.
   Laissez l’option Row key Cassandra type à Default, car sa valeur devient automatiquement Int32.


9. Si nécessaire, les champs Key start et Key end vous permettent de définir l’écart de lignes. Le champ Key limit vous permet de spécifier le nombre de lignes à l’intérieur de l’écart de lignes à lire. De la même manière, les champs Columns range start et Columns range end vous permettent de définir l’écart de colonnes dans la famille de colonnes. Le champ Columns range limit vous permet de spécifier le nombre de colonnes dans l’écart de colonnes à lire.

**Afficher les informations qui vous intéressent**

**Procédure**

1. Double-cliquez sur le composant tLogRow pour ouvrir sa vue Component.
2. Dans la zone Mode, sélectionnez Table (print values in cells of a table) afin de visualiser les données sous forme de tableau.

**Fermer la connexion Cassandra**

**Procédure**

1. Double-cliquez sur le composant tCassandraClose pour afficher sa vue Component.

2. Sélectionnez la connexion à fermer, dans la liste Component List.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl+S pour sauvegarder votre Job.
2. Exécutez le Job en appuyant sur la touche F6 ou en cliquant sur le bouton Run dans la vue Run. Les informations personnelles des trois employés s’affichent dans la console.
<table>
<thead>
<tr>
<th>Display_Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
tCassandraOutput

Ce composant écrit des données dans une famille de colonne d’un Keyspace Cassandra ou en supprime.
Le composant tCassandraOutput reçoit des données du composant précédent et les écrit dans Cassandra.

Propriétés du tCassandraOutput Standard

Ces propriétés sont utilisées pour configurer le tCassandraOutput s’exécutant dans le framework de Jobs Standard.
Le composant tCassandraOutput Standard appartient aux familles Big Data et Databases.
Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Property type       | Peut être Built-In ou Repository.  
|                     | **Built-In** : propriétés utilisées ponctuellement.  
|                     | **Repository** : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.  
| Use existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
| DB Version | Sélectionnez la version de Cassandra que vous utilisez.  
| API type | Cette liste déroulante s’affiche uniquement lorsque vous avez sélectionné la version 2.0 de Cassandra dans la liste DB version. Dans cette liste API type, vous pouvez sélectionner Datastax pour utiliser CQL 3 (Cassandra Query Language) avec Cassandra ou sélectionnez Hector pour utiliser CQL 2. 
| Host | Saisissez le nom de l’hôte ou l’adresse IP du serveur Cassandra.  
| Port | Saisissez le numéro du port d’écoute du serveur Cassandra.  

Notez que l’API Hector est dépréciée depuis la version 2.0 de Cassandra, mais est toujours disponible pour une utilisation dans le Studio, à des fins de flexibilité de version du langage de requête à utiliser avec Cassandra 2.0.0.  
Avec l’évolution des commandes CQL, les paramètres à configurer dans la vue Basic settings varient.
tCassandraOutput

Required authentication

Cochez cette case afin de saisir vos identifiants de
connexion à Cassandra.
Cette case apparaît si vous ne cochez pas la case Use
existing connection.

Username

Saisissez dans ce champ l'identifiant d'authentification à
Cassandra.

Password

Saisissez dans ce champ le mot de passe d'authentifica
tion à Cassandra.
Pour saisir le mot de passe, cliquez sur le bouton [...] à
côté du champ Password, puis, dans la boîte de dialogue
qui s'ouvre, saisissez le mot de passe entre guillemets
doubles et cliquez sur OK afin de sauvegarder les
paramètres.

Use SSL

Cochez cette case pour activer la connexion chiffrée SSL
ou TLS.
Utilisez le composant tSetKeystore dans le même Job
afin de spécifier les informations de chiffrement.
Pour plus d'informations concernant le tSetKeystore,
consultez tSetKeystore à la page 3745.

Keyspace

Saisissez le nom de l'espace de clé (Keyspace) dans
lequel vous souhaitez écrire les données.

Action on keyspace

Sélectionnez l'opération que vous souhaitez effectuer
sur le Keyspace à utiliser :
•

None : aucune opération n'est effectuée.

•

Drop and create keyspace : le Keyspace est supprimé
et créé à nouveau.

•

Create keyspace : le Keyspace n'existe pas et est
créé.

•

Create keyspace if not exists: le Keyspace est créé s'il
n'existe pas.

•

Drop keyspace if exists and create : le Keyspace est
supprimé s'il existe déjà et créé à nouveau.

Column family

Saisissez le nom de la famille de colonnes dans laquelle
vous souhaitez écrire les données.

Action on column family

Sélectionnez l'opération à effectuer sur la famille de
colonnes à utiliser :
•

None : aucune opération n'est effectuée.

•

Drop and create column family : la famille de
colonnes est supprimée et recréée.

•

Create column family : la famille de colonnes
n'existe pas et est créée.

•

Create column family if not exists : la famille de
colonnes est créée si elle n'existe pas.

455


- **Drop column family if exists and create**: la famille de colonnes est supprimée si elle existe déjà et recréée.

**Action on data**

Sur les données de la table définie, vous pouvez effectuer les actions suivantes :

- **Upsert**: insérer les colonnes si elles n'existent pas ou mettre à jour les colonnes existantes.
- **Insert**: insérer les colonnes si elles n'existent pas. Cette action met également à jour les colonnes existantes.
- **Update**: mettre à jour les colonnes existantes ou ajoute les colonnes qui n'existent pas. Cette action ne supporte pas le type de données Cassandra **Counter**.
- **Delete**: supprimer les colonnes correspondantes du flux d’entrée.

Notez que la liste des actions varient selon l’API, **Hector** ou **Datastax**, que vous utilisez. Lorsque l’API sélectionnée est **Datastax**, plus d’options sont disponibles.

Pour plus d’options avancées, utilisez la vue **Advanced settings**.

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

<table>
<thead>
<tr>
<th>Sync columns</th>
<th>Cliquez sur ce bouton afin de récupérer le schéma du composant précédent dans le Job.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die on error</td>
<td>Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien <strong>Row &gt; Reject</strong>.</td>
</tr>
<tr>
<td>Column family type</td>
<td><strong>Standard</strong> : La famille de colonnes est de type standard. <strong>Super</strong> : La famille de colonnes est de type super.</td>
</tr>
</tbody>
</table>

Sélectionnez l’opération que vous souhaitez effectuer sur l’espace de clé :

- **None** : Aucune opération n’est effectuée.
- **Drop and create keyspace** : Le keyspace est supprimé et créé à nouveau.
- **Create keyspace** : Le keyspace n’existe pas et est créé.
- **Create keyspace if not exists** : Le keyspace est créé s’il n’existe pas.
- **Drop keyspace if exists and create** : Le keyspace est supprimé s’il existe déjà et est recréé.

Sélectionnez la colonne de clé de ligne dans la liste.

Cochez cette case pour inclure la clé de ligne dans les colonnes.

Sélectionnez la super colonne, dans la liste.

Cette liste déroulante apparaît si vous sélectionnez **Super** dans la liste **Column family type**.

Cochez cette case pour inclure les super colonnes dans les colonnes standards.

Cochez cette case pour supprimer la ligne.

Cette case apparaît uniquement lorsque vous sélectionnez **Delete** dans la liste déroulante **Action on data**.

Sélectionnez ou personnalisez les colonnes à supprimer.

Cochez cette case pour supprimer des super colonnes.
Cette case apparaît uniquement si vous cochez la case **Delete Row**.

### Advanced settings

<table>
<thead>
<tr>
<th>Batch Size</th>
<th>Nombre de lignes dans chaque lot traité. Lorsque vous utilisez l'API <strong>Datastax</strong>, cette fonctionnalité s’affiche uniquement lorsque vous avez coché la case <strong>Use unlogged batch</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Global Variables

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie et nécessite un lien d’entrée. <strong>Use unlogged batch</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Insert if not exists</strong></td>
<td>Cochez cette case pour insérer des lignes. Cette insertion de lignes s’effectue uniquement si elles n’existent pas dans la table cible. Cette fonctionnalité est disponible uniquement lorsque l’action <strong>Insert</strong> est sélectionnée.</td>
</tr>
<tr>
<td><strong>Delete if exists</strong></td>
<td>Cochez cette case pour supprimer de la table cible les lignes ayant le même enregistrement dans le flux entrant.</td>
</tr>
</tbody>
</table>
Cette fonctionnalité est disponible uniquement lorsque l’action **Delete** est sélectionnée.

### Use TTL

Cochez cette case pour écrire les données du TTL dans la table cible. Dans la liste de colonnes affichée, sélectionnez la colonne à utiliser comme colonne du TTL. Le type de cette colonne doit être **Int**.

Cette fonctionnalité est disponible uniquement lorsque l’action **Insert ou Update** est sélectionnée.

### Use Timestamp

Cochez cette case pour écrire les données de date et heure dans la table cible. Dans la liste de colonnes affichée, sélectionnez la colonne à utiliser pour stocker les données de date et heure. Le type de cette colonne doit être **BigInt**.

Cette fonctionnalité est disponible avec les actions suivantes : **Insert, Update et Delete**.

### IF condition

Ajoutez la condition à rencontrer pour que l’action **Update ou Delete** s’exécute. Cette condition vous permet d’être plus précis quant à la colonne à mettre à jour ou à supprimer.

### Special assignment operation

Renseignez cette table afin de construire les commandes avancées SET de Cassandra, pour rendre l’action **Update** plus spécifique. Par exemple, ajouter un enregistrement au début ou à une position particulière d’une colonne donnée.

Dans la colonne **Update column** de la table, vous devez sélectionnez la colonne à mettre à jour puis sélectionner les opérations à utiliser, depuis la colonne **Operation**.

Les opérations suivantes sont disponibles :

- **Append** : ajoute les enregistrements entrants à la fin de la colonne à mettre à jour. Les types de données gérés par Cassandra sont **Counter, List, Set et Map**.

- **Prepend** : ajoute les enregistrements entrants au début de la colonne à mettre à jour. Le seul type de données géré par Cassandra est **List**.

- **Remove** : supprime les enregistrements de la table cible lorsque les mêmes enregistrements existent dans le flux entrant. Les types de données gérés par Cassandra sont **Counter, List, Set et Map**.

- **Assign based on position/key** : ajoute des enregistrements à une position précise de la colonne à mettre à jour. Les types de données gérés par Cassandra sont **List et Map**.

Une fois cette opération sélectionnée, la colonne **Map key/list position** devient modifiable. Dans cette colonne, vous devez sélectionner la colonne à utiliser en tant que référence afin de localiser la position à mettre à jour.

<table>
<thead>
<tr>
<th>Scénario associé</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pour un scénario utilisant le tCassandraOutput, consultez Scénario : Gestion des données avec Cassandra à la page 447.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Row key in the List type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez la colonne à utiliser pour construire la clause WHERE de Cassandra pour effectuer l’action Update ou Delete sur les lignes sélectionnées. Les colonnes à utiliser dans cette table doivent provenir de l’ensemble des colonnes de clé primaire de la table Cassandra.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Delete collection column based on position/key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez la colonne à utiliser comme référence afin de localiser la ou les ligne(s) à supprimer. Cette fonctionnalité est disponible uniquement lorsque l’action Delete est sélectionnée.</td>
</tr>
</tbody>
</table>
tCassandraOutputBulk

Ce composant prépare une table SSTable de grande taille et de la traiter selon vos besoins avant de charger cette table SSTable dans une famille de colonne d’un Keyspace Cassandra.

Les composants tCassandraOutputBulk et tCassandraBulkExec sont généralement utilisés ensemble en tant que parties d’un processus de deux étapes. Dans la première étape, une table SSTable est générée. Dans la seconde étape, cette table SSTable est écrite dans Cassandra. Ces deux étapes sont fusionnées dans le composant tCassandraOutputBulkExec, détaillé dans une section séparée. L’avantage d’utiliser deux composant séparés réside dans le fait que les données peuvent être transformées avant d’être chargées dans Cassandra.

Le composant tCassandraOutputBulk reçoit des données du composant précédent et crée localement une table SSTable.

Propriétés du tCassandraOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tCassandraOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tCassandraOutputBulk Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

**Table type**

Sélectionnez le type du modèle de données à utiliser pour la table à créer. Le type peut être **CQL** (plus précisément CQL3) ou **non-CQL** (L'API basée Thrift héritée de Cassandra avant CQL3).

Cette liste déroulante est disponible uniquement lorsque la version sélectionnée dans la liste **DB version** est Cassandra 2.0.0. Pour les versions de Cassandra supérieures à 2.0.0, CQL devient le seul modèle utilisé par ce composant et la liste n’est plus disponible.

**DB Version**

Sélectionnez la version de Cassandra que vous utilisez.

**Host**

Saisissez le nom de l’hôte ou l’adresse IP du serveur Cassandra.

**Port**

Saisissez le numéro du port d’écoute du serveur Cassandra.

**Required authentication**

Cochez cette case pour fournir les informations d’authentification à Cassandra.

⚠️ Avertissement :

Cette case ne s’affiche pas si vous sélectionnez Cassandra 1.1.2 dans la liste **DB Version**.

**Username**

Renseignez ce champ en saisissant l’identifiant d’authentification à Cassandra.

**Password**

Renseignez ce champ en saisissant le mot de passe d’authentification à Cassandra.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Use configuration file**

Cochez cette case et, dans le champ qui s’affiche, saisissez le chemin d’accès ou parcourez votre système jusqu’au fichier principal de configuration pour Cassandra, **cassandra.yaml**.

Ainsi, ce composant peut importer et utiliser directement la configuration du fichier **cassandra.yaml**, qui peut contenir des propriétés avancées de Cassandra, comme les propriétés relatives au chiffrement SSL.
Lorsque vous devez exécuter votre Job dans différents environnements Cassandra, cette fonctionnalité permet à votre Job de passer facilement d’une configuration à une autre.

Pour plus d’informations concernant ce fichier `cassandra.yaml`, consultez Cassandra configuration (en anglais).

### Keyspace
Saisissez le nom du Keyspace dans lequel écrire la table SSTable.

### Column family
Saisissez le nom de la famille de colonnes dans laquelle écrire la table SSTable.

### Partitioner
Sélectionnez le partitionneur déterminant comment les données sont partitionnées à travers le cluster Cassandra.

- **Random**: partitionneur par défaut dans Cassandra 1.1 et versions antérieures.
- **Murmur3**: partitionneur par défaut dans Cassandra 1.2.
- **Order preserving**: non recommandé car il suppose que les clés sont des chaînes de caractères en UTF8.


### Schema statement
Saisissez l’instruction pour définir le schéma de la famille de colonnes à utiliser ou à créer à la volée.

- Cette instruction est une instruction préparée Cassandra, stockant localement les résultats de requête dans le répertoire SSTable directory défini via ce composant, avant de les envoyer au serveur. Pour plus d’informations concernant les instructions préparées, consultez [Prepared statements](http://wiki.apache.org/cassandra/Prepared%20statements) (en anglais).
- Une famille de colonnes Cassandra est un conteneur d’une collection de lignes d’enregistrements partageant un même type. Son schéma doit contenir la même colonne que le schéma du composant défini, c’est-à-dire que les noms de colonnes doivent être identiques dans les deux schémas.

Un exemple d’instruction pour ce schéma est fourni dans le champ **Schema statement** :

```sql
create table ks.tb (id int, name text, birthday timestamp, primary key(id, birthday)) with clustering order by (birthday desc)
```

Cette instruction va appeler une famille de colonnes nommée `tb` et contenant les colonnes `id`, `name` et `birthday` sous le Keyspace `ks`.

Pour plus d'informations concernant une famille de colonnes, consultez [Standard column family](http://wiki.apache.org/cassandra/Standard%20column%20family) (en anglais).
Ce champ est disponible uniquement lorsque la version sélectionnée de votre base de données Cassandra est supérieure à 2.0.0. Lorsque votre version est 2.0.0, ce champ est disponible lorsque vous avez sélectionné **CQL** dans la liste **Table type**.

### Insert statement

Saisissez l'instruction pour écrire les données du flux d'entrée dans les colonnes de la famille à utiliser.

Cette instruction est une instruction préparée Cassandra, stockant localement les résultats de requête dans le répertoire **SSTable directory** défini via ce composant, avant de les envoyer au serveur. Pour plus d'informations concernant les instructions préparées, consultez **Prepared statements** (en anglais).

Un exemple de cette instruction d’insertion est fourni dans le champ **Insert statement** :

```java
insert into ks.tb (id, name, birthday) values (?, ?, ?)
```

Cette instruction va écrire des données dans les colonnes **id**, **name** et **birthday**, respectivement d’une famille de colonnes nommée **tb** dans le Keyspace **ks**. Les points d’interrogation dans l’instruction sont les marqueurs des variables liées pour les trois colonnes. Pour plus d’informations concernant les variables liées et leur utilisation, consultez **Bound parameters** et **Parameterized queries** (en anglais).

Ce champ est disponible uniquement lorsque la version sélectionnée de votre base de données Cassandra est supérieure à 2.0.0. Lorsque votre version est 2.0.0, ce champ est disponible lorsque vous avez sélectionné **CQL** dans la liste **Table type**.

### Column name comparator

Sélectionnez le type de données pour le nom des colonnes, utilisé pour trier les colonnes. Cette liste n’est pas disponible lorsque le modèle de données à utiliser est CQL.3.


### SSTable directory

Spécifiez le répertoire local pour la table SSTable. Notez que le chemin d’accès complet à la table SSTable comprend le répertoire local à la suite du Keyspace spécifié et du nom de la famille de colonnes.

Par exemple, si vous configurez le répertoire local à `/home/talend/sstable` et que vous spécifiez **testk** comme nom du Keyspace et **testc** comme nom de la famille de colonnes, le chemin d’accès complet à la table SSTable est `/home/talend/sstable/testk/testc/`.

### Buffer size

Spécifiez la taille à partir de laquelle écrire la table SSTable dans Cassandra.
**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant nécessite un lien d’entrée. |

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tCassandraOutputBulkExec**

Ce composant dédié permet un gain de performance pendant les opérations d’Insert dans une famille de colonnes dans un Keyspace Cassandra.

Les composants `tCassandraOutputBulk` et `tCassandraBulkExec` sont généralement utilisés ensemble pour écrire des données en sortie dans une table SSTable et écrire la table SSTable dans Cassandra, au cours d’un processus de deux étapes. Ces deux étapes sont fusionnées dans le composant `tCassandraOutputBulkExec`.

Le `tCassandraOutputBulkExec` reçoit des données du composant précédent, crée une table SSTable et écrit cette table SSTable dans Cassandra.

**Propriétés du tCassandraOutputBulkExec Standard**

Ces propriétés sont utilisées pour configurer le `tCassandraOutputBulkExec` s’exécutant dans le framework de Jobs Standard.

Le composant `tCassandraOutputBulkExec Standard` appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de **Talend**.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
• **View schema** : sélectionnez cette option afin de voir le schéma.
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**. |

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

**Table type**

Sélectionnez le type du modèle de données à utiliser pour la table à créer. Le type peut être **CQL** (plus précisément CQL3) ou **non-CQL** (L’API basée Thrift héritée de Cassandra avant CQL3).

Cette liste déroulante est disponible uniquement lorsque la version sélectionnée dans la liste **DB version** est Cassandra 2.0.0. Pour les versions de Cassandra supérieures à 2.0.0, CQL devient le seul modèle utilisé par ce composant et la liste n’est plus disponible.

**DB Version**

Sélectionnez la version de Cassandra que vous utilisez.

⚠️ **Avertissement** :
- Pour Cassandra 1.1.2, seules les visites en local sont supportées sous Linux.
- Pour Cassandra 1.2.2, assurez-vous que le serveur et le client fonctionnent sous le même système d'exploitation.
- Cassandra 2.0.0 fonctionne uniquement avec la JVM 1.7.

**Host**

Saisissez le nom de l’hôte ou l’adresse IP du serveur Cassandra.

**Port**

Saisissez le numéro du port d’écoute du serveur Cassandra.

**Required authentication**

Cochez cette case afin de fournir les informations d’authentification pour Cassandra.

⚠️ **Avertissement** :
Cette case n’apparaît pas si vous sélectionnez Cassandra 1.1.2 dans la liste **DB Version**.

**Username**

Saisissez dans ce champ le nom d’utilisateur pour l’authentification à Cassandra.

**Password**

Saisissez dans ce champ le mot de passe pour l’authentification à Cassandra.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets.
doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

| **Keyspace** | Saisissez le nom du Keyspace dans lequel écrire la table SSTable. |
| **Column family** | Saisissez le nom de la famille de colonnes dans laquelle écrire la table SSTable. |
| **Partitioner** | Sélectionnez le partitionneur déterminant comment les données sont partitionnées à travers le cluster Cassandra.  
  - Random : partitionneur par défaut dans Cassandra 1.1 et versions antérieures.  
  - Murmur3 : partitionneur par défaut dans Cassandra 1.2.  
  - Order preserving : non recommandé car il suppose que les clés sont des chaînes de caractères en UTF8.  
| **Schema statement** | Saisissez l’instruction pour définir le schéma de la famille de colonnes à utiliser ou à créer à la volée.  
  - Cette instruction est une instruction préparée Cassandra, stockant localement les résultats de requête dans le répertoire SSTable directory défini via ce composant, avant de les envoyer au serveur. Pour plus d’informations concernant les instructions préparées, consultez [Prepared statements](http://wiki.apache.org/cassandra/PreparedStatements) (en anglais).  
  - Une famille de colonnes Cassandra est un conteneur d’une collection de lignes d’enregistrements partageant un même type. Son schéma doit contenir la même colonne que le schéma du composant défini, c’est-à-dire que les noms de colonnes doivent être identiques dans les deux schémas.  
  Un exemple d’instruction pour ce schéma est fourni dans le champ **Schema statement** :  
  ```sql  
  create table ks.tb (id int, name text, birthday timestamp, primary key(id, birthday)) with clustering order by (birthday desc)  
  ```  
  Cette instruction va appeler une famille de colonnes nommée *tb* et contenant les colonnes *id, name* et *birthday* sous le Keyspace *ks*.  
  Pour plus d’informations concernant une famille de colonnes, consultez [Standard column family](http://wiki.apache.org/cassandra/StandardColumnFamily) (en anglais).  
  Ce champ est disponible uniquement lorsque la version sélectionnée de votre base de données Cassandra est supérieure à 2.0.0. Lorsque votre version est 2.0.0, ce
Saisissez l'instruction pour écrire les données du flux d'entrée dans les colonnes de la famille à utiliser.

Cette instruction est une instruction préparée Cassandra, stockant localement les résultats de requête dans le répertoire `SSTable directory` défini via ce composant, avant de les envoyer au serveur. Pour plus d'informations concernant les instructions préparées, consultez [Prepared statements](en anglais).

Un exemple de cette instruction d'insertion est fourni dans le champ **Insert statement** :

```
insert into ks.tb (id, name, birthday) values (?, ?, ?)
```

Cette instruction va écrire des données dans les colonnes `id`, `name`, et `birthday`, respectivement d'une famille de colonnes nommée `tb` dans le Keyspace `ks`. Les points d’interrogation dans l’instruction sont les marqueurs des variables liées pour les trois colonnes. Pour plus d’informations concernant les variables liées et leur utilisation, consultez [Bound parameters](en anglais) et [Parameterized queries](en anglais).

Ce champ est disponible uniquement lorsque la version sélectionnée de votre base de données Cassandra est supérieure à 2.0.0. Lorsque votre version est 2.0.0, ce champ est disponible lorsque vous avez sélectionné **CQL** dans la liste **Table type**.

Sélectionnez le type de données pour le nom des colonnes, utilisé pour trier les colonnes.


Spécifiez le répertoire local de la table SSTable à charger dans Cassandra. Notez que le chemin d’accès complet à la table SSTable comprend le répertoire local à la suite du Keyspace spécifié et du nom de la famille de colonnes.

Par exemple, si vous configurez le répertoire local à `/home/talend/sstable` et que vous spécifiez `testk` comme nom du Keyspace et `testc` comme nom de la famille de colonnes, le chemin d’accès complet à la table SSTable est `/home/talend/sstable/testk/testc/`

Spécifiez la taille à partir de laquelle écrire la table SSTable dans Cassandra.

Cochez cette case pour collecter les données de log au niveau Job ainsi qu’au niveau des composants.
Global Variables

Global Variables

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé lorsqu’aucune transformation n’est requise sur les données à charger dans la base de données.</th>
</tr>
</thead>
</table>

| Limitation         | L’exécution de ce composant provoque la fin du Job. |

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
Ce composant agit sur la structure même de la base de données ou sur les données, selon la nature de la requête et de la base de données.

Le composant tCassandraRow est le composant spécifique pour les requêtes, dans cette base de données. Il exécute des requêtes en langage Cassandra Query Language (CQL) dans la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

**Propriétés du tCassandraRow Standard**

Ces propriétés sont utilisées pour configurer le tCassandraRow s’exécutant dans le framework de Jobs Standard.

Le composant tCassandraRow Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être Built-In ou Repository. Built-In : propriétés utilisées ponctuellement. Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
<tr>
<td><strong>Use existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>DB Version</strong></td>
<td>Sélectionnez la version de Cassandra que vous utilisez.</td>
</tr>
<tr>
<td><strong>API type</strong></td>
<td>Cette liste déroulante s’affiche uniquement lorsque vous avez sélectionné la version 2.0 de Cassandra dans la liste DB version. Dans cette liste API type, vous pouvez sélectionner Datastax pour utiliser CQL 3 (Cassandra Query Language) avec Cassandra ou sélectionnez Hector pour utiliser CQL 2. Notez que l’API Hector est dépréciée depuis la version 2.0 de Cassandra, mais est toujours disponible pour une utilisation dans le Studio, à des fins de flexibilité de version du langage de requête à utiliser avec Cassandra 2.0.0.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Saisissez le nom de l’hôte ou l’adresse IP du serveur Cassandra.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Saisissez le numéro du port d’écoute du serveur Cassandra.</td>
</tr>
<tr>
<td><strong>Required Authentication</strong></td>
<td>Cochez cette case afin de renseigner vos informations d’authentification à Cassandra.</td>
</tr>
</tbody>
</table>
Cette case apparaît uniquement si vous ne cochez pas la case **Use existing connection**.

| **Username** | Renseignez dans ce champ votre identifiant d'authentification à Cassandra. |
| **Password** | Saisissez dans ce champ le mot de passe d'authentification à Cassandra.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Keyspace** | Saisissez le nom du keyspace sur lequel vous souhaitez exécuter les commandes CQL. |
| **Column family** | Nom de la famille de la colonne. |

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Query**

Saisissez la commande CQL à exécuter.

Par défaut, la requête n’est pas sensible à la casse. Cela signifie, que, lors de l’exécution, le nom des colonnes saisis dans la requête sont toujours pris en minuscules. Si vous voulez rendre la requête sensible à la casse, mettez les noms de colonnes entre guillemets doubles.

**Die on error**

La case est décochée par défaut. Décochez la case afin d’ignorer la ligne d’erreur et finir le processus de lignes sans erreur.
Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

Global Variables


Utilisation

| Limitation | Les commandes select ne sont pas supportées. |

Scénarios associés

Pour des scénarios associés, consultez :
- Scénario : Supprimer et re-générer un index de table MySQL à la page 2700.
- Scénario 2 : Utiliser l’instance PreparedStatement pour faire une requête sur des données à la page 2702.
tCentricCRMInput

Ce composant permet d’extraire les données d’une base de données Centric CRM à l’aide d’une requête.

Le tCentricCRMInput se connecte à un module de la base de données CentricCRM via le service Web adéquat.

**Propriétés du tCentricCRMInput Standard**

Ces propriétés sont utilisées pour configurer le tCentricCRMInput s’exécutant dans le framework de Jobs Standard.

Le composant tCentricCRMInput Standard appartient à la famille Business.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>CentricCRM URL</th>
<th>Saisissez l’URL du service Web permettant de se connecter à la base de données CentricCRM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module</td>
<td>Sélectionnez le module adéquat dans la liste.</td>
</tr>
<tr>
<td>Server</td>
<td>Saisissez l’adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>UserID et Password</td>
<td>Saisissez les informations d’authentification de l’utilisateur au service Web.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

**Schema et Edit Schema**

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la</td>
</tr>
</tbody>
</table>
métadonnée du schéma dans la fenêtre [Repository Content].
Dans ce composant, le schéma est relatif au Module sélectionné.

**Query condition**
Saisissez la requête permettant de sélectionner les données à extraire.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu’au niveau de chaque composant. |

**Global Variables**

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire. |

**Limitation**
Ce composant requiert l’installation des fichiers .jar liés.

**Scénario associé**
Aucun scénario n’est disponible pour la version Standard de ce composant.
tCentricCRMOutput

Ce composant écrit des données dans une base de données CentricCRM.
Le tCentricCRMOutput écrit dans un module de la base de données CentricCRM via le service Web adéquat.

**Propriétés du tCentricCRMOutput Standard**

Ces propriétés sont utilisées pour configurer le tCentricCRMOutput s’exécutant dans le framework de Jobs Standard.
Le composant tCentricCRMOutput Standard appartient à la famille Business.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>CentricCRM URL</th>
<th>Saisissez l’URL du service Web permettant de se connecter à la base de données CentricCRM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module</td>
<td>Sélectionnez le module adéquat dans la liste.</td>
</tr>
<tr>
<td>Server</td>
<td>Saisissez l’adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>UserID et Password</td>
<td>Saisissez les informations d’authentification de l’utilisateur au service Web.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Action</td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données du module CentricCRM :</td>
</tr>
<tr>
<td></td>
<td><strong>Insert</strong> : insérer les données.</td>
</tr>
<tr>
<td></td>
<td><strong>Update</strong> : mettre les données à jour.</td>
</tr>
<tr>
<td></td>
<td><strong>Delete</strong> : supprimer les données.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
</tbody>
</table>
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu’au niveau de chaque composant.</th>
</tr>
</thead>
</table>

### Global Variables

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le [Guide utilisateur du Studio Talend](#). |

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est utilisé comme composant de sortie. Il nécessite un composant d’entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Ce composant requiert l’installation des fichiers .jar liés.</td>
</tr>
</tbody>
</table>

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
### tChangeFileEncoding

Ce composant transforme l'encodage des caractères d'un fichier donné, et génère un nouveau fichier en fonction de l'encodage des caractères transformés.

Le composant tChangeFileEncoding permet de modifier l'encodage d'un fichier donné.

### Propriétés du tChangeFileEncoding Standard

Ces propriétés sont utilisées pour configurer le tChangeFileEncoding s'exécutant dans le framework de Jobs Standard.

Le composant tChangeFileEncoding Standard appartient aux familles Data Quality et File.

Le composant de ce framework est toujours disponible.

#### Basic settings

| Use Custom Input Encoding | Cochez cette case pour personnaliser le type d'encodage des caractères. Lorsque cette case est cochée, une liste des types d'encodage d'entrée apparaît, vous permettant soit de sélectionner un type d'encodage d'entrée parmi les encodings proposés, soit de spécifier vous-même un type d'encodage d'entrée en sélectionnant CUSTOM.
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoding</td>
<td>A partir de cette liste de types d'encodage, vous pouvez sélectionnez l'une des options proposées, ou personnaliser l'encodage des caractères en sélectionnant CUSTOM, et spécifier un type d'encodage de caractères.</td>
</tr>
<tr>
<td>Input File Name</td>
<td>Chemin d'accès au fichier d'entrée.</td>
</tr>
<tr>
<td>Output File Name</td>
<td>Chemin d'accès au fichier de sortie.</td>
</tr>
</tbody>
</table>

#### Advanced settings

<table>
<thead>
<tr>
<th>Create directory if does not exist</th>
<th>Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez la case afin de collecter les données de log au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

#### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>EXISTE : résultat spécifiant si un fichier existe ou non. Cette variable est une variable Flow et retourne un booléen.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FILENAME : nom du fichier traité. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | Ce composant doit être utilisé en standalone. |

## Scénario : Transformer l’encodage des caractères d’un fichier

Ce scénario Java décrit un Job très simple permettant de transformer l’encodage des caractères d’un fichier texte, et de générer un nouveau fichier en tenant compte du nouvel encodage des caractères.

### Procédure

1. A partir de la **Palette**, cliquez et déposez un composant **tChangeFileEncoding** dans l’espace de modélisation graphique.

2. Double-cliquez sur le composant **tChangeFileEncoding** pour afficher l’onglet **Basic settings** de sa vue **Component**.

```
<table>
<thead>
<tr>
<th>tChangeFileEncoding_1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic settings</strong></td>
</tr>
<tr>
<td>Use Custom Input Encoding</td>
</tr>
<tr>
<td>Input File Name</td>
</tr>
<tr>
<td>Output File Name</td>
</tr>
<tr>
<td>Encoding</td>
</tr>
</tbody>
</table>
```
3. Cochez la case **Use Custom Input Encoding**. Sélectionnez le type d’encodage “GB2312” dans la liste déroulante **Encoding**.
4. Dans le champ **Input File Name**, parcourez ou saisissez le chemin d’accès au fichier d’entrée.
5. Dans le champ **Output File Name**, parcourez ou saisissez le chemin d’accès au fichier de sortie.
6. Sélectionnez **CUSTOM** dans la liste déroulante **Encoding**, et saisissez “UTF-16” dans le champ texte.
7. Sauvegardez votre Job et appuyez sur **F6** pour l’exécuter.

**Résultats**
Le type d’encodage du fichier *in.txt* est transformé, et *out.txt* est généré avec un type d’encodage UTF-16.
tChronometerStart

Ce composant fonctionne comme un chronomètre : le composant commence à calculer soit le temps de traitement d’un ou plusieurs sous-Jobs du Job principal, soit le temps de traitement d’une partie d’un sous-Job.

Le composant tChronometerStart démarre l’opération de mesure du temps d’exécution d’un sous-Job.

Propriétés du tChronometerStart Standard

Ces propriétés sont utilisées pour configurer le tChronometerStart s’exécutant dans le framework de Jobs Standard.

Le composant tChronometerStart Standard appartient à la famille Logs & Errors.

Le composant de ce framework est toujours disponible.

Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</th>
</tr>
</thead>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>STARTTIME : heure de départ utilisée pour calculer le temps de traitement du (des) sous-job(s). Cette variable est une variable Flow et retourne un long.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur générée par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Utilisation

| Règle d’utilisation | Vous pouvez utiliser le tChronometerStart comme composant de début ou au milieu d’un Job. Il peut précéder une ou plusieurs tâches de traitement dans un sous-Job. Il peut également précéder un ou plusieurs sous-Jobs dans le Job principal. |
Scénario associé

Pour un scénario associé, consultez Scénario : Mesurer la durée de traitement d’un sous-Job ou d’une partie d’un sous-Job à la page 484.
tChronometerStop

Ce composant fonctionne comme un chronomètre : il affiche le temps d’exécution total et arrête le calcul du temps de traitement soit d’un ou de plusieurs sous-Jobs dans le Job principal, soit d’une partie d’un sous-Job.

Le composant tChronometerStop mesure le temps d’exécution d’un sous-Job.

Propriétés du tChronometerStop Standard

Ces propriétés sont utilisées pour configurer le tChronometerStop s’exécutant dans le framework de Jobs Standard.

Le composant tChronometerStop Standard appartient à la famille Logs & Errors.

Le composant de ce framework est toujours disponible.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Basic settings**

| Since options | Cochez l’une des deux cases pour choisir le point de départ du processus de mesure :  
**Since the beginning** : arrête le calcul du temps dont la mesure a commencé au début d’un sous-Job.  
**Since a tChronometerStart** : arrête le calcul du temps dont la mesure a commencé à partir d’un composant tChronometerStart utilisé sur le flux de données du sous-Job. |
| Display duration in console | Cochez cette case pour afficher les informations d’exécution du sous-Job sur la console. |
| Display component name | Cochez cette case pour afficher le nom des composants sur la console. |
| Caption | Entrez un texte, pour permettre d’identifier votre sous-Job par exemple. |
| Display human readable duration | Cochez cette case pour afficher les informations d’exécution du sous-Job en unités de temps exploitables. |

**Global Variables**

| Global Variables | STOPTIME : heure de fin utilisée pour calculer le temps de traitement du (des) sous-job(s). Cette variable est une variable Flow et retourne un long. |
| **DURATION** : temps de traitement du (des) sous-job(s). Cette variable est une variable **Flow** et retourne un long.
| **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

| Règle d’utilisation | Ne peut être utilisé comme composant de début. |

### Scénario : Mesurer la durée de traitement d’un sous-Job ou d’une partie d’un sous-Job

Ce scénario est un sous-Job, qui, dans une séquence, effectue les opérations suivantes :

- génère 1 000 000 lignes de noms et de prénoms ;
- regroupe les noms et les prénoms correspondants ;
- stocke les données de sortie dans un fichier délimité ;
- mesure la durée d’ensemble du sous-Job ;
- mesure la durée de l’opération de remplacement des noms,
- affiche les informations relatives au temps de traitement dans la console de log **Run**.

Pour mesurer le temps d’exécution du sous-Job :

- A partir de la **Palette**, cliquez-déposez les composants suivants dans l’éditeur graphique : le **tRowGenerator**, le **tMap**, le **tFileOutputDelimited** et le **tChronometerStop**.
- Reliez les trois premiers composants à l’aide de liens de type **Main Row**.

1. **Remarque** :
   Lorsque vous reliez le **tMap** au **tFileOutputDelimited**, vous serez invités à donner un nom à la table en sortie. Le nom utilisé dans cet exemple est "new_order".
• Reliez le **tFileOutputDelimited** au **tChronometerStop** à l'aide d'un lien de type **OnComponentOk**.

• Sélectionnez le **tRowGenerator** puis cliquez sur la vue **Component** pour paramétrer ses propriétés de base, définies par défaut, dans l'onglet **Basic settings**.

• Cliquez sur le bouton [...] du champ **Edit schema** pour définir le schéma du **tRowGenerator**. Cliquez deux fois sur le bouton [+] pour ajouter deux colonnes : **First_Name** et **Last_Name**. Cliquez sur **OK** afin de fermer la boîte de dialogue.

• Cliquez sur le bouton [...] du champ **RowGenerator Editor** pour ouvrir l'éditeur et définir les données à générer.

• Dans le champ **Number of Rows for RowGenerator** de l'éditeur [RowGenerator Editor], précisez le nombre de lignes à générer puis cliquez sur **OK**. L'éditeur [RowGenerator Editor] se ferme.

• Une boîte de dialogue s'ouvre et vous demande si vous souhaitez propager les changements. Cliquez sur **Yes** (Oui).

• Double-cliquez sur le composant **tMap** pour ouvrir son éditeur. L'éditeur du **tMap** affiche le schéma du composant **tRowGenerator**.
• Dans l’onglet Schema editor de l’éditeur du tMap, cliquez sur le bouton [+] de la table de sortie pour ajouter deux lignes puis configurez-les.

• Dans l’éditeur du tMap, glissez la ligne First_Name de la table d’entrée dans la ligne Last_Name de la table de sortie et glissez la ligne Last_Name dans la ligne First_Name de la table de sortie.

• Cliquez sur Apply pour enregistrer les modifications.

• A nouveau, une boîte de dialogue s’ouvre et vous demande si vous souhaitez propager les changements. Cliquez sur Yes.

• Cliquez sur OK pour fermer l’éditeur.
• Sélectionnez le tFileOutputDelimited et cliquez sur la vue Component pour paramétrer ses propriétés de base dans l'onglet Basic settings :

<table>
<thead>
<tr>
<th>Basic settings</th>
<th>Property Type</th>
<th>Built-In</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced settings</td>
<td>Use Output Stream</td>
<td>✔️</td>
</tr>
<tr>
<td>Dynamic settings</td>
<td>FileName</td>
<td>D:/Output/out.csv</td>
</tr>
<tr>
<td>View</td>
<td>Row Separator</td>
<td>&quot;n&quot;</td>
</tr>
<tr>
<td></td>
<td>Apprend</td>
<td>✔️</td>
</tr>
<tr>
<td>Documentation</td>
<td>Schema</td>
<td>Built-In</td>
</tr>
</tbody>
</table>

• Sélectionnez le tChronometerStop et cliquez sur la vue Component pour paramétrer ses propriétés de base dans l'onglet Basic settings.

• Dans la zone Since options, cochez l’option Since the beginning pour mesurer la durée d’exécution du sous-Job entier.

<table>
<thead>
<tr>
<th>Basic settings</th>
<th>Since options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced settings</td>
<td>Since the beginning</td>
</tr>
<tr>
<td>Dynamics settings</td>
<td>Since a tChronometerStart</td>
</tr>
<tr>
<td>View</td>
<td>Display duration in console</td>
</tr>
<tr>
<td></td>
<td>Display component name</td>
</tr>
<tr>
<td></td>
<td>Display human readable duration</td>
</tr>
</tbody>
</table>

• Cochez/décochez les autres cases en fonction de vos besoins. Dans ce scénario, l’objectif est d’afficher dans la console la durée d’exécution du sous-Job, précédée du nom du composant.

• Si nécessaire, saisissez du texte de le champ Caption.

• Enregistrez votre Job et appuyez sur F6 pour l’exécuter.

```
Starting job tChronometerStop at 11:40 05/03/2010.
[statistics] connecting to socket on port 3399
[statistics] connected
[ tChronometerStop_1 ] 2seconds duration of the subjob
2938 milliseconds
[statistics] disconnected
Job tChronometerStop ended at 11:40 05/03/2010. [exit code=0]
```

Remarque :

Vous pouvez aussi mesurer la durée du sous-Job en plaçant un tChronometerStop en dessous du tRowGenerator, et en les connectant via un lien OnSubJobOk.
tCloudStart

Ce composant démarre des instances sur Amazon EC2 (Amazon Elastic Compute Cloud).
Le composant tCloudStart accède au fournisseur Cloud à utiliser (Amazon EC2) et lance des instances, qui sont des serveurs virtuels dans ce Cloud. Si une instance à lancer n’existe pas, le tCloudStart la crée.

Propriétés du tCloudStart Standard

Ces propriétés sont utilisées pour configurer le tCloudStart s’exécutant dans le framework de Jobs Standard.
Le composant tCloudStart Standard appartient à la famille Cloud.
Le composant de ce framework est toujours disponible.

**Basic settings**


Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud provider</td>
<td>Sélectionnez le fournisseur Cloud à utiliser.</td>
</tr>
<tr>
<td>Image</td>
<td>Saisissez le nom de l’image de la machine Amazon (Amazon Machine Image) à utiliser pour lancer une instance. L’AMI définit la configuration de base de cette instance.</td>
</tr>
</tbody>
</table>
| Region et Zone | Saisissez la région et la zone à utiliser comme emplacement géographique où lancer une instance.

La syntaxe utilisée pour exprimer un lieu est prédéfinie par Amazon. Par exemple, *us-east-1* représente la région *US East (Northern Virginia)* et *us-east-1a* représente l’une es zones de disponibilité dans cette région. Pour plus d’informations à propos des régions disponibles pour Amazon, consultez la documentation Amazon concernant les régions et les endpoints ainsi que la FAQ Amazon des régions et des zones de disponibilité. |
| Instance name | Saisissez le nom de l’instance à lancer. Par exemple, vous pouvez saisir *Talend*.

Notez que toute lettre en majuscule sera convertie en minuscule. |
### Instance count
Saisissez le nombre d’instances à lancer. Au moment de l’exécution, le nom spécifié dans le champ *Instance name*, par exemple *Talend*, est utilisé comme première partie du nom de chaque instance. Des lettres et des chiffres sont ajoutés au hasard afin de compléter chaque nom.

### Instance type
 Sélectionnez le type d’instance(s) à lancer. Chaque type est prédéfini par Amazon et définit la performance de chaque instance à lancer.

Cette liste déroulante présente le nom de l’API pour chaque type d’instance. Pour plus d’informations, consultez la documentation Amazon concernant les types d’instance.

### Proceed with a Key pair
 Cochez cette case pour utiliser la paire de clés Amazon (Key Pair) afin de vous connecter à Amazon EC2. Une fois la case cochée, une liste déroulante apparaît. Vous pouvez sélectionner :

- **Use an existing Key Pair** pour saisir le nom de cette paire de clés dans le champ à côté de la liste déroulante. Si nécessaire, Amazon vous demande de trouver et d’utiliser cette paire de clés lors de l’exécution.

- **Create a Key Pair** pour saisir le nom de la nouvelle paire de clés dans le champ à côté de la liste déroulante et définir l’emplacement où stocker cette paire de clés, dans la vue *Advanced settings*.

### Security group
Ajoutez des lignes dans cette table et saisissez les noms des groupes de sécurité auxquels vous devez assigner l’instance ou les instances à lancer. Les groupes de sécurité configurés dans cette table doivent exister dans votre Amazon EC2.

Un groupe de sécurité applique des règles spécifiques sur le trafic entrant dans les instances assignées au groupe, comme les ports à utiliser. Pour plus d’informations concernant les groupes de sécurité, consultez la documentation Amazon concernant les groupes de sécurité.

Notez qu’une instance peut être assignée à un groupe en paramétrant son nom de groupe de sécurité ou son nom de paire de clés à `jclouds#$<group_name>`, où `<group_name>` identifie le groupe auquel appartient l’instance. De cette manière, vous pouvez modifier le statut de toutes les instances ou des instances en cours d’exécution dans un groupe, simultanément, à l’aide du composant *tCloudStop*.

### Advanced settings

#### Key Pair folder
Parcourez votre système, ou saisissez le chemin d’accès au dossier à utiliser pour stocker la paire de clés créée.
Ce champ s’affiche lorsque vous sélectionnez **Creating a Key Pair** dans l’onglet **Basic settings**.

**Volumes**

Ajoutez des lignes et définissez le(s) volume(s) à créer pour les instances à lancer, en plus des volumes prédéfinis et alloués par votre Amazon EC2.

Les paramètres à configurer dans cette table ont la même valeur que les paramètres utilisés Amazon pour décrire un volume.

Si vous devez supprimer automatiquement un volume supplémentaire après la fin de son instance relative, cochez la case dans la colonne **Delete on termination**.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau des composants.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NODE_GROUP : nom de l’instance. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NODES : instances en cours d’exécution. Cette variable est une variable <strong>After</strong> et retourne un object.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

### Utilisation

**Règle d’utilisation**

Ce composant peut être utilisé en standalone afin de lancer une instance de Amazon EC2. Vous pouvez utiliser ce composant pour démarrer l’instance sur laquelle déployer des Jobs.

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tCloudStop

Ce composant modifie le statut d'une instance lancée sur Amazon EC2 (Amazon Elastic Compute Cloud).

Le composant tCloudStart accède au fournisseur Cloud à utiliser (Amazon EC2) et suspend, reprend ou termine une ou des instance(s) données(s).

Propriétés du tCloudStop Standard

Ces propriétés sont utilisées pour configurer le tCloudStop s'exécutant dans le framework de Jobs Standard.

Le composant tCloudStop Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

| Access key et Secret key | Saisissez ou collez la clé d'accès et la clé secrète requises par Amazon pour authentifier vos requêtes dans ses services Web. Ces informations d'accès sont générées dans l'onglet Security Credential de la page de votre compte Amazon.

Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.

| Cloud provider | Sélectionnez le fournisseur Cloud à utiliser.

| Action | Sélectionnez l’action que doit effectuer le tCloudStop afin de modifier le statut d’une instance donnée. L’action peut être :

- Suspend
- Resume
- Terminate

Notez que si vous terminez une instance, cette instance est supprimée, alors que vous pouvez toujours reprendre une instance suspendue.

| Predicate | Sélectionnez l’instance ou les instances dont vous souhaitez modifier le statut. Les options sont les suivantes :

- Running instances : le statut de toutes les instances en cours est modifié.
- Instances in a specific group : le statut des instances d’un groupe d’instances spécifique est modifié. Vous devez saisir le nom de ce groupe dans le champ Group name.
• **Running instances in a specific group** : le statut des instances en cours d’un groupe d’instances spécifiques est modifié. Vous devez saisir le nom de ce groupe dans le champ *Group name*.

• **Instance with predefined id** : le statut d’une instance donnée est modifié. Vous devez saisir l’ID de cette instance dans le champ *Id*. Vous pouvez trouver cet ID dans votre Amazon EC2.

Un groupe d’instances se compose d’instances utilisant le même nom d’instance, défini dans le champ *Instance name* du composant *tCloudStart*.

**Group name**

Saisissez le nom du groupe dans lequel vous souhaitez modifier le statut des instances données dont le nom du groupe de sécurité ou le nom de la paire de clés est configuré à jclouds#<group_name> dans le composant *tCloudStart*, où <group_name> identifie le groupe auquel appartient l’instance.

Ce champ est disponible uniquement lorsque vous sélectionnez *Instances in a specific group* ou *Running instances in a specific group* dans la liste *Predicate*.

**Id**

Saisissez l’ID de l’instance dont vous devez changer le statut. Le champ apparaît lorsque vous sélectionnez *Instance with predefined id* dans la liste *Predicate*.

---

**Advanced settings**

- **tStatCatcher Statistics**
  
  Cochez cette case pour collecter les données de log au niveau des composants.

---

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

  Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

  Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

  Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

---

**Utilisation**

- **Règle d’utilisation**

  Ce composant fonctionne en standalone pour changer le statut d’instances données d’Amazon EC2. Vous pouvez
utiliser ce composant afin de suspendre, reprendre ou terminer une ou des instance(s) où sont déployés des Jobs.

Ce composant est généralement utilisé avec le tCloudStart afin de changer le statut des instances lancées par celui-ci.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tCombinedSQLAggregate

Ce composant fournit un ensemble de matrices basées sur des valeurs ou sur des calculs.

Le composant tCombinedSQLAggregate collecte les valeurs de données d’une ou plusieurs colonnes d’une table dans des buts statistiques. Ce composant exécute en temps réel des transformations de données dans le SGBD lui-même.

Propriétés du tCombinedSQLAggregate Standard

Ces propriétés sont utilisées pour configurer le tCombinedSQLAggregate s’exécutant dans le framework de Jobs Standard.

Le composant tCombinedSQLAggregate Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].  
Cliquez sur le bouton Sync columns afin de récupérer le schéma du composant précédent. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend .</td>
</tr>
<tr>
<td><strong>Group by</strong></td>
<td>Définit les ensembles d’agrégation, dont les valeurs sont utilisées pour les calculs.</td>
</tr>
</tbody>
</table>
### Output Column
Sélectionnez le libellé de colonne dans la liste fournie, basée sur la structure de schéma que vous avez définie. Vous pouvez ajouter autant de colonnes de sortie que vous le souhaitez afin d'affiner les agrégations.

### Input Column
Sélectionnez les libellés des colonnes d’entrée afin de les faire correspondre au contenu attendu des colonnes de sortie, dans le cas où vous souhaitez que les libellés du schéma de sortie soient différents du schéma d’entrée.

### Operations
Sélectionnez le type d’opération et la valeur à utiliser pour le calcul et le champ de sortie.

### Output Column
Sélectionnez le champ de destination dans la liste.

### Function
Sélectionnez l’une des opérations suivantes à effectuer sur les données : count, min, max, avg, sum, first, last, distinct et count (distinct).

### Input column
Sélectionnez la colonne d’entrée à partir de laquelle vous souhaitez collecter les valeurs à agréger.

### Advanced settings

#### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

### Global Variables

#### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable Flow et retourne un entier.</td>
</tr>
<tr>
<td>QUERY</td>
<td>Requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
Utilisation

| Règle d’utilisation | Ce composant est utilisé en tant que composant intermédiaire avec d’autres composants de bases de données, en particulier les composants de connexion et de commit. |

Scénario : Filtre et agréger des colonnes d’une table directement sur le SGBD

Le scénario suivant crée un Job ouvrant une connexion à une base de données MySQL et :

- alimente une table de base de données avec les données source,
- crée une table de sortie pour les données filtrées,
- instancie en partie les schémas d’une table de base de données (pour filtrer les colonnes),
- filtre deux colonnes dans la même table afin de ne garder que les données correspondant aux deux conditions de filtre,
- collecte les données des colonnes filtrées, regroupées par valeur(s) spécifique(s) et écrit les données agrégées dans une table cible de base de données.
Ajouter et relier les composants

Procédure

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique :
   tMysqlConnection, tFixedFlowInput, tMysqlOutput, tCreateTable, tCombinedSQLInput,
   tCombinedSQLFilter, tCombinedSQLAggregate, tCombinedSQLOutput, tMysqlCommit, tMysqlInput
   et tLogRow.
2. Reliez le composant tMysqlConnection au tFixedFlowInput à l’aide d’un lien Trigger > OnSubjobOk.
3. Procédez de la même manière afin de relier le tFixedFlowInput au tCreateTable, le tCreateTable au tCombinedSQLInput, le tCombinedSQLInput au tMysqlCommit et le tMysqlCommit au tMysqlInput.
4. Reliez le composant tFixedFlowInput au tMysqlOutput à l’aide d’un lien Row > Main.
5. Reliez le composant tCombinedSQLInput au tCombinedSQLFilter à l’aide d’un lien Row > Combine.
6. Procédez de la même manière afin de relier le tCombinedSQLFilter au tCombinedSQLAggregate et le tCombinedSQLAggregate au tCombinedSQLOutput.
7. Reliez le composant tMysqlInput au tLogRow à l’aide d’un lien Row > Main.

## Configurer les composants

Le schéma défini dans le composant tCombinedSQLInput peut être différent de celui de la table source, puisque vous pouvez instancier uniquement les colonnes souhaitées de la table source. Le tCombinedSQLInput joue également un rôle de filtre des colonnes.

Dans ce scénario, la table de la base de données source contient sept colonnes : *id, first_name, last_name, city, state, date_of_birth*, ainsi que *salary*. Le composant tCombinedSQLInput instancie uniquement les quatre colonnes nécessaires à l’agrégation : *id, state, date_of_birth* et *salary* de la table source.

### Ouvrir une connection MySQL

#### Procédure

1. **Lancez MySQL Workbench** et démarrez une connexion locale sur le port 3306.
2. **Créez un nouveau schéma que vous nommerez test**.
3. **De retour dans l’espace de modélisation graphique, double-cliquez sur le composant tMysqlConnection pour afficher sa vue Component et définir ses propriétés de base.**
4. **Dans la vue Basic settings, configurez manuellement les informations de connexion à la base de données ou sélectionnez Repository dans la liste Property Type si votre connexion à la base de données a déjà été configurée et stockée sous le nœud Metadata de la vue Repository.**

Pour plus d’informations concernant le stockage des métadonnées dans le Repository, consultez le Guide utilisateur du Studio Talend.
Insérer les données dans la table source

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le composant **tFixedFlowInput** pour afficher sa vue **Component** et définir ses propriétés de base.

2. Dans la vue **Basic settings**, saisissez 500 dans le champ **Number of rows**.

3. Dans ce scénario, la table de la base de données source contient sept colonnes : *id, first_name, last_name, city, state, date_of_birth*, ainsi que *salary*.

   Cliquez sur le bouton [...] à côté du champ **Edit schema** pour définir la structure suivante pour les données.
4. Cliquez sur l’icône représentant une disquette afin de sauvegarder le schéma pour une utilisation ultérieure.

5. Dans la fenêtre **Select folder**, sélectionnez **default** et cliquez sur **OK**.

6. Saisissez le nom que vous avez choisi pour votre schéma et cliquez sur **Finish**.

7. Cliquez sur **OK**.

8. La première colonne de la table **Values** reflète automatiquement la structure de données que vous avez précédemment configurée.


10. Dans l’espace de modélisation graphique, double-cliquez sur le composant **tMysqlOutput** pour afficher sa vue **Component** et définir ses propriétés de base.

Le schéma de sortie sera automatiquement le même que celui du composant qui le précède, **tFixedFlowInput** dans ce cas.

**Créer la table de sortie dans la base de données**

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le composant **tCreateTable** pour afficher sa vue **Component** et définir ses propriétés de base.
2. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour définir la structure suivante pour les données.

Le schéma que vous spécifiez à cette étape doit déjà correspondre aux différentes opérations d’agrégation que vont subir les données source.

**Extraire et filtrer les données**

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le composant **tCombinedSQLInput** pour afficher sa vue **Component** et définir ses propriétés de base.

2. Dans le champ **Table**, saisissez *employees* comme nom pour la table source.

3. Dans le champ **Schema**, choisissez **Repository** dans la liste déroulante et cliquez sur le bouton [...] à côté du champ vide pour charger le schéma enregistré lors de la configuration du **tFixedFlowInput**.
4. Dans la fenêtre Repository Content, développez Generic schemas et choisissez votre schéma.
5. Cliquez sur le bouton [...] à côté du champ Edit schema.
6. Sélectionnez View schema et dans la première colonne de la table, décochez les cases first_name, last_name et city.

Filtrer et agréger les données source

Procédure

1. Dans l’espace de modélisation graphique, double-cliquez sur le composant tCombinedSQLFilter afin d’afficher sa vue Component et définir ses propriétés de base.

2. Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent, ou configurez manuellement le schéma en sélectionnant Built-in dans la liste Schema et en cliquant sur le bouton [...] à côté du champ Edit schema.

Lorsque vous définissez la structure des données du composant tCombinedSQLFilter, le nom des colonnes s’affichent automatiquement dans la liste Input column dans la table Conditions.

Dans ce scénario, le composant tCombinedSQLFilter instancie quatre colonnes : id, state, date_of_birth et salary.

3. Dans la table Conditions, configurez les paramètres d’entrée, les opérateurs et les valeurs attendues pour extraire les enregistrements correspondant aux critères spécifiés.

Cliquez deux fois sur le bouton [+] sous la table Conditions et dans Input column, sélectionnez state et date_of_birth dans la liste déroulante.

Dans ce scénario, le composant tCombinedSQLFilter filtre les colonnes state et date_of_birth dans la table source afin d’extraire uniquement les employés nés après le 19 octobre 1960, habitant dans les États Utah, Ohio et Iowa.

4. Pour la colonne state, choisissez l’opérateur IN depuis la liste déroulante, puis saisissez (Utah,'Ohio','Iowa') en tant que valeur.

5. Pour la colonne date_of_birth, choisissez l’opérateur > depuis la liste déroulante, puis saisissez (1960-10-19) en tant que valeur.

6. Sélectionnez And dans la liste Logical operator between conditions afin d’appliquer les deux conditions à la fois. Vous pouvez également personnaliser les conditions en cochant la case Use custom SQL et en modifiant les conditions dans le champ de texte.

7. Dans l’espace de modélisation graphique, double-cliquez sur le composant tCombinedSQLAggregate afin d’afficher sa vue Component et définir ses propriétés de base.
8. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour définir la structure suivante :

Le composant **tCombinedSQLAggregate** instancie quatre colonnes : *id*, *state*, *date_of_birth* et *salary*, provenant du composant précédent.

9. Le tableau **Group by** vous permet de définir les ensembles de données à traiter à partir d’une colonne définie, *State* dans cet exemple.

   Dans le tableau **Group by**, cliquez sur le bouton [*+] pour ajouter une ligne.

10. Dans la liste déroulante **Output column**, sélectionnez *State*. Cette colonne sera utilisée pour stocker les données filtrées de la colonne *State*.

11. Le tableau **Operations** vous permet de configurer le type d’agrégation à effectuer. La liste disponible dans la colonne **Output column** dépend du schéma que vous souhaitez écrire en sortie (via le composant **tCombinedSQLOutput**). Dans ce scénario, le but est de regrouper les employés selon l’État dans lequel ils vivent, de calculer le nombre d’employés par État, les salaires moyen/minimum/maximum ainsi que de retourner la date de naissance de l’employé le plus âgé et de l’employé le plus jeune de chaque État.
12. Dans le tableau **Operations**, cliquez sur le bouton **[+]** pour ajouter une ou plusieurs ligne(s) puis cliquez dans la liste **Output column** afin de sélectionner la colonne de sortie qui contiendra les données calculées.

13. Cliquez dans la ligne **Function** et sélectionnez l’opération à effectuer.

**Écrire les données en sortie dans MySQL**

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le t**CombinedSQLOutput** afin d’afficher sa vue **Component** et définir ses propriétés de base.

2. Dans la liste **Database type**, sélectionnez la base de données correspondante.

3. Dans la liste **Component List**, sélectionnez le composant de connexion correspondant si plus d’un composant de connexion est utilisé.

4. Dans le champ **Table**, saisissez le nom de la table cible qui stockera les résultats agrégés, dans ce cas, empl_by_state.

Le composant t**CombinedSQLOutput** nécessite qu’une table de sortie soit préalablement créée dans la base de données pour fonctionner. C’est pourquoi la table empl_by_state a été créée plus tôt dans le scénario.

Dans cet exemple, il n’est pas nécessaire de renseigner le champ **Schema**, puisque la base de données utilisée est Oracle.

5. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

Dans ce scénario, le composant t**CombinedSQLOutput** instancie sept colonnes provenant du composant précédent dans le Job (t**CombinedSQLAggregate**): state, empl_count, avg_salary, min_salary, max_salary, oldest_empl et youngest_empl.

**Committer les données dans la base de données**

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le t**CombinedSQLCommit** afin d’afficher sa vue **Component** et définir ses propriétés de base.

2. Dans la liste **Component List**, sélectionnez le composant de connexion à la base de données correspondant si plusieurs ont été utilisés.

3. Décochez la case **Close Connection**.
Récupérer les données filtrées et agrégées

Procédure

1. Dans l'espace de modélisation graphique, double-cliquez sur le tMysqlInput afin d'afficher sa vue Component et définir ses propriétés de base.

2. Cochez la case Use an existing connection.

3. Cliquez sur le bouton [...] à côté du champ Edit schema pour définir la structure suivante :

4. Dans le champ Table Name, saisissez empl_by_state et dans le champ Query, entrez select * from empl_by_state.

5. Dans l'espace de modélisation graphique, double-cliquez sur le tLogRow afin d'afficher sa vue Component et définir ses propriétés de base.
6. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent et sélectionnez le mode **Table (print values in cells of a table)**.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Sauvegardez votre Job et appuyez sur **F6** pour l’exécuter.
2. L’onglet **Run** s’ouvre, dans lequel vous pouvez observer le résultat de l’exécution du Job.
3. Les données traitées sont récupérées par le composant **tLogRow** et sont visibles sous forme de table.
   Des lignes sont insérées dans une table composée de sept colonnes, *empl_by_state*, dans la base de données. La table affiche, par État défini, le nombre d’employés, les salaires moyen/minimum/maximum et retourne la date de naissance de l’employé le plus âgé et de l’employé le plus jeune de chaque État.

```
Starting job MyJob at 20:10 23-04-2016
[statistics] connecting to socket on port 3903
[statistics] connected

<table>
<thead>
<tr>
<th>state</th>
<th>empl_count</th>
<th>avg_salary</th>
<th>min_salary</th>
<th>max_salary</th>
<th>oldestempl</th>
<th>youngestempl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ohio</td>
<td>9</td>
<td>5670</td>
<td>3332323333</td>
<td>707</td>
<td>9744</td>
<td>9730</td>
</tr>
<tr>
<td>Utah</td>
<td>8</td>
<td>44777</td>
<td>2500000000</td>
<td>440</td>
<td>9154</td>
<td>9110</td>
</tr>
</tbody>
</table>

[statistics] disconnected
Job MyJob ended at 20:10 23-04-2016. [exit code=0]
```
tCombinedSQLFilter

Ce composant filtre des données en réorganisant, supprimant ou ajoutant des colonnes à partir de la table source et de filtrer la source de données à l’aide de conditions de filtre.

Le composant tCombinedSQLFilter vous permet de modifier le schéma de la table source via le mapping des colonnes et de définir un filtre sur cette table. Ce composant peut donc filtrer des colonnes et des lignes en même temps. Il filtre en temps réel les données dans le SGBD lui-même.

Propriétés du tCombinedSQLFilter Standard

Ces propriétés sont utilisées pour configurer le tCombinedSQLFilter s’exécutant dans le framework de Jobs Standard.

Le composant tCombinedSQLFilter Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical operator between conditions</td>
<td>Sélectionnez l’opérateur logique à ajouter entre les conditions de filtre définies dans le tableau Conditions. Les deux opérateurs disponibles sont : Or et And.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Cliquez sur le bouton [+] pour ajouter autant de conditions que voulu. Les conditions sont exécutées les unes après les autres pour chaque ligne.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Input Column</td>
<td>Sélectionnez la colonne à filtrer.</td>
</tr>
<tr>
<td>Operator</td>
<td>Sélectionnez le type de clause WHERE : =, &lt; &gt;, &gt;, &lt;, &gt;&gt;, &lt;=, LIKE, NOT IN, et EXIST IN.</td>
</tr>
<tr>
<td>Value</td>
<td>Saisissez les valeurs à utiliser dans la clause WHERE.</td>
</tr>
<tr>
<td>Negate</td>
<td>Cochez cette case pour activer la condition contraire à celle définie.</td>
</tr>
<tr>
<td>Use custom SQL</td>
<td>Cochez cette case afin de personnaliser une clause WHERE en éditant le code dans le champ SQL Condition.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable Flow et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

| Règle d’utilisation | Ce composant est utilisé en tant que composant intermédiaire avec d’autres composants de bases de données, en particulier les composants de connexion et de commit. |
Scénario associé

Pour un scénario associé, consultez Scénario : Filtre et agréger des colonnes d’une table directement sur le SGBD à la page 496.
**tCombinedSQLInput**

Ce composant extrait des champ d’une table de base de données selon la définition de son schéma.

Il passe ensuite la liste des champs au composant suivant via un lien de type **Combine**. Le schéma du tCombinedSQLInput peut être différent de celui de la table source de la base de données mais l’ordre des colonnes doit être le même.

Le composant tCombinedSQLInput extrait des champs d’une table de base de données à partir de son schéma. Ce composant possède également des fonctionnalités de filtre puisque son schéma peut être différent de celui de la table de la base de données.

**Propriétés du tCombinedSQLInput Standard**

Ces propriétés sont utilisées pour configurer le tCombinedSQLInput s’exécutant dans le framework de Jobs Standard.

Le composant tCombinedSQLInput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Table</th>
<th>Saisissez le nom de la table source de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Saisissez le nom du schéma de la table source. Renseignez ce champ si vous utilisez Oracle.</td>
</tr>
</tbody>
</table>

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

| Built-in | Le schéma sera créé et conservé pour ce composant seulement. Voir également le du Guide utilisateur du **Studio Talend**. |

**Add additional columns**

Cette option vous permet d’effectuer des actions sur les colonnes, à l’exception des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.

**Name** : Saisissez le nom de la colonne du schéma devant être modifiée.

**SQL expression** : Saisissez l’expression SQL à exécuter pour modifier les données dans la colonne correspondante.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

**Global Variables**

| Global Variables | NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable Flow et retourne un entier. |
| | QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères. |
| | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
| | Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |
| | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
| | Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| Règle d’utilisation | Ce composant est utilisé en tant que composant intermédiaire avec d’autres composants de bases de données, en particulier les composants de connexion et de commit. |
Scénario associé

Pour un scénario associé, consultez Scénario : Filtre et agréger des colonnes d’une table directement sur le SGBD à la page 496.
tCombinedSQLOutput

Ce composant insère des enregistrements du flux entrant dans une table de base de données existante.

Propriétés du tCombinedSQLOutput Standard

Ces propriétés sont utilisées pour configurer le tCombinedSQLOutput s’exécutant dans le framework de Jobs Standard.

Le composant tCombinedSQLOutput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Database Type</th>
<th>Sélectionnez le type de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant de connexion à la base de données correspondant si plus d’un est utilisé dans le Job.</td>
</tr>
<tr>
<td>Table</td>
<td>Saisissez le nom de la table cible de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Saisissez le nom du schéma de la table cible. Renseignez ce champ si vous utilisez Oracle.</td>
</tr>
</tbody>
</table>

Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton Sync columns afin de récupérer le schéma du composant précédent.
### Built-in
Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository

### Action on data
 Sélectionnez **INSERT** dans la liste afin d’insérer les enregistrements du flux d’entrée dans la table cible de la base de données.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>NB_LINE</strong> : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <strong>Flow</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>QUERY</strong> : requête traitée. Cette variable est une variable <strong>Flow</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéDER à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

| Règle d’utilisation | Ce composant est utilisé en tant que composant intermédiaire avec d’autres composants de bases de données, en particulier les composants de connexion et de commit. |
**Scénario associé**

Pour un scénario associé, consultez Scénario : Filtre et agréger des colonnes d’une table directement sur le SGBD à la page 496.
tContextDump

Ce composant copie la configuration du contexte du Job courant dans un fichier plat, une table de base de données, etc, pouvant être réutilisé(e) par le tContextLoad.

Utilisé avec le tContextLoad, ce composant permet d’appliquer simplement la configuration de contexte d’un Job à un autre.

tContextDump passe la configuration du contexte du Job courant au composant suivant.

**Propriétés du tContextDump Standard**

Ces propriétés sont utilisées pour configurer le tContextDump s’exécutant dans le framework de Jobs Standard.

Le composant tContextDump Standard appartient à la famille Misc.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit les champs qui sont traités et passés au composant suivant.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Remarque :</td>
</tr>
<tr>
<td></td>
<td>Le schéma du tContextDump est en lecture seule et se compose de deux colonnes, Key et Value, qui correspondent au nom et à la valeur du paramètre du contexte du Job.</td>
</tr>
</tbody>
</table>

| Hide Password | Cochez cette case pour masquer la valeur du mot de passe du paramètre de contexte, c’est-à-dire pour afficher des * pour la valeur des paramètres de contexte dont le Type est Password. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | Ce composant est un composant d’entrée et passe la configuration du contexte du Job à un fichier, une table d’une base de données, etc. |

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tContextLoad**

Ce composant charge un contexte à partir d'un flux entrant.

Ce composant effectue aussi deux contrôles. Il prévient lorsque les paramètres définis dans le flux entrant ne sont pas définis dans le contexte. Il avertit également lorsque la valeur d'un contexte n'est pas initialisée dans le flux entrant. Notez que ces avertissements ne bloquent pas le traitement.

Le composant tContextLoad modifie dynamiquement les valeurs du contexte actif.

**Propriétés du tContextLoad Standard**

Ces propriétés sont utilisées pour configurer le tContextLoad s'exécutant dans le framework de Jobs Standard.

Le composant tContextLoad Standard appartient à la famille Misc.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champ qui sont traités et passés au composant suivant. Le schéma du composant tContextLoad doit être composé de deux colonnes, l'une contenant le nom du paramètre et la deuxième contenant la valeur du paramètre à charger.</th>
</tr>
</thead>
<tbody>
<tr>
<td>If a variable loaded, but not in the context</td>
<td>Si une variable est chargée mais n'apparaît pas dans le contexte, choisissez comment doit s'afficher la notification. Sous forme d'erreur (Error), de warning (warning), ou d'information (info).</td>
</tr>
<tr>
<td>If a variable in the context, but not loaded</td>
<td>Si une variable apparaît dans le contexte mais n'est pas chargée, choisissez comment doit s'afficher la notification. Sous forme d'erreur (Error), de warning (warning), ou d'information (info).</td>
</tr>
<tr>
<td>Print operations</td>
<td>Cochez cette case pour afficher les paramètres de contexte dans la vue Run.</td>
</tr>
<tr>
<td>Disable errors</td>
<td>Cochez cette case pour ne pas afficher d'erreur.</td>
</tr>
<tr>
<td>Disable warnings</td>
<td>Cochez cette case pour ne pas afficher de warning.</td>
</tr>
<tr>
<td>Disable infos</td>
<td>Cochez cette case pour ne pas afficher d'information.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et les lignes contenant les erreurs seront ignorées.</td>
</tr>
</tbody>
</table>
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier. |
|                 | KEY_NOT_INCONTEXT : les variables sont chargées mais n’apparaissent pas dans le contexte. Cette variable est une variable After et retourne une chaîne de caractères. |
|                 | KEY_NOT_LOADED : les variables ne sont pas chargées mais apparaissent dans le contexte. Cette variable est une variable After et retourne une chaîne de caractères. |
|                 | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | Ce composant s’appuie sur le flux de données pour charger les valeurs de contexte à utiliser. Ainsi, il requiert un composant d’entrée et ne peut pas être un composant de début. |

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin d’activer ou désactiver dynamiquement l’option Print operations au moment de l’exécution. |

Lorsqu’un paramètre dynamique est configuré, l’option Print operations devient inaccessible dans la vue Basic settings.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les
**Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement**

Le Job décrit dans ce scénario est composé de deux sous-jobs. Le premier permet de charger dynamiquement des paramètres de contexte à partir de deux fichiers texte. Le second utilise ces paramètres chargés afin de se connecter à deux bases de données différentes et d’afficher le contenu d’une table de base de données existante pour chacune de ces bases. Les paramètres de contexte du Job permettent de décider à quelle base de données se connecter et de choisir d’afficher ou non les paramètres de contexte définis dans la console à l’exécution.

**Déposer et relier les composants**

**Procédure**

1. Déposez, depuis la Palette, un **tFileInputDelimited** et un **tContextLoad** dans l’espace de modélisation graphique. Reliez-les à l’aide d’un lien **Row > Main** afin de créer le premier sous-job.
2. Déposez un **tMysqlInput** et un **tLogRow** dans l’espace de modélisation graphique. Reliez-les à l’aide d’un lien **Row > Main** afin de créer le second sous-job.
3. Reliez les deux sous-jobs à l’aide d’un lien **Trigger > OnSubjobOk**.

**Préparer les contextes et les variables de contexte**

**Procédure**

1. Créez deux fichiers délimités correspondant aux deux contextes de ce scénario, à savoir les accès aux deux bases de données et nommez-les respectivement **test_connection.txt** et **prod_connection.txt**. Ces fichiers contiennent les informations de connexion aux bases de données,
respectivement pour le test et pour la production. Chaque fichier est composé de deux colonnes, contenant le nom des paramètres et leurs valeurs correspondantes. Voici un exemple :

```
host;localhost
port;3306
database;test
username;root
password;talend
```

2. Sélectionnez la vue **Contexts** puis cliquez sept fois sur le bouton [+] pour ajouter sept lignes à la table afin de définir les paramètres suivants :

- *host*, de type *String*
- *port*, de type *String*
- *database*, de type *String*
- *username*, de type *String*
- *password*, type *Password*
- *filename*, de type *File*
- *printOperations*, de type *Boolean*.

Notez que les paramètres *host*, *port*, *database*, *username* et *password* correspondent au nom des paramètres dans les fichiers délimités et sont utilisés pour configurer la connexion à la base de données souhaitée. Le paramètre *filename* est utilisé pour définir le fichier délimité à lire au moment de l’exécution du Job et le paramètre *printOperations* est utilisé pour décider d’afficher ou non les paramètres de contexte définis par le `tContextLoad` dans la console.

3. Cliquez sur l’onglet **Contexts** et cliquez sur le bouton [+] situé en haut à droite du panneau afin d’ouvrir la boîte de dialogue [Configure Contexts].

4. Sélectionnez le contexte par défaut, cliquez sur le bouton **Edit** et renommez le contexte en *Test*.

5. Cliquez sur **New** afin d’ajouter un nouveau contexte nommé *Production*. Cliquez ensuite sur **OK** pour fermer la boîte de dialogue.
6. De retour dans l'onglet **Contexts**, définissez la valeur de la variable *filename* sous chaque contexte, en cliquant dans le champ **Value** respectif et en parcourant votre système jusqu'au fichier délimité.

7. Cochez la case **Prompt** près du champ **Value** de la variable *filename*, pour que les deux contextes affichent les champs **Prompt** et saisissez le message à afficher lors de l'exécution.

8. Pour la variable *printOperations*, cliquez dans le champ **Value** sous le contexte *Production* et sélectionnez **false** dans la liste. Cliquez dans le champ **Value** sous le contexte *Test* et sélectionnez **true** dans la liste. Cochez la case **Prompt** sous chaque contexte et saisissez le message à afficher lors de l'exécution.

---

**Configurer les composants**

**Procédure**

1. Dans le champ **File name/Stream** de la vue **Basic settings** du tFileInputDelimited, saisissez le nom de la variable de contexte correspondante définie précédemment : `context.filename`.

2. Définissez manuellement (**Built-in**) le schéma du fichier. Il contient deux colonnes : *key* et *value*.

3. Acceptez la propagation du schéma défini au composant suivant (tContextLoad).

4. Dans la vue **Dynamic settings** du tContextLoad, cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez `context.printOperations` afin d'utiliser la variable de contexte *printOperations* définie précédemment. Notez que la case **Print Operations** de la vue **Basic settings** devient grisée et inaccessible.
5. Double-cliquez sur le **tMysqlInput** afin d’ouvrir sa vue **Basic settings**.

6. Dans les champs **Host**, **Port**, **Database**, **Username** et **Password**, saisissez les variables stockées dans les fichiers délimités et définies dans la vue **Contexts** : `context.host`, `context.port`, `context.database`, `context.username`, et `context.password` respectivement dans cet exemple. Dans le champ **Table name**, saisissez le nom de la table de base de données à partir de laquelle lire les données, **customers** pour les deux bases de données de cet exemple.

7. Dans le champ **Schema**, saisissez les informations sur le schéma. Si vous avez stocké le schéma dans le nœud **Metadata** du **Repository**, sélectionnez-le après avoir sélectionné l’option **Repository**. Dans cet exemple, le schéma des deux bases de données est composé de quatre colonnes : `id` (de type **INT** et d’une longueur de 2 caractères), `firstname` (de type **VARCHAR** et d’une longueur de 15 caractères), `lastName` (de type **VARCHAR** et d’une longueur de 15 caractères) et `city` (de type **VARCHAR** et d’une longueur de 15 caractères).

8. Dans le champ **Query**, saisissez la requête SQL à exécuter sur la table de base de données spécifiée. Dans cet exemple, cliquez sur **Guess Query** afin de récupérer toutes les colonnes de la table. Elles s’affichent dans la vue **Run** grâce au **tLogRow**.
9. Dans la vue **Basic settings** du tLogRow, sélectionnez l’option **Table** afin d’afficher les enregistrements sous forme de tableau.

**Exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** afin de sauvegarder le Job. Appuyez sur **F6** afin d’exécuter le Job avec le contexte par défaut, *Test* dans ce scénario.

Une boîte de dialogue s’affiche, vous demandant de spécifier le fichier délimité à lire et de décider d’afficher ou non les paramètres de contexte définis dans la console.

Si vous le souhaitez, vous pouvez spécifier un fichier autre que celui par défaut. Vous pouvez également décocher la case **Show loaded variables** si vous ne souhaitez pas voir les variables de contexte dans la console. Afin d’exécuter le Job avec les paramètres par défaut, cliquez sur **OK**.
Les paramètres de contexte et le contenu de la table de base de données correspondant au contexte Test s’affichent dans la console Run.

2. Sélectionnez à présent le contexte Production et appuyez sur F6 afin d’exécuter le Job à nouveau. Lorsque la boîte de dialogue s’affiche, cliquez sur OK pour exécuter le Job avec les paramètres par défaut.

Le contenu de la table de base de données correspondant au contexte Test s’affiche dans la console Run. Comme la variable printOperations a la valeur false, les paramètres de contexte définis ne s’affichent pas dans la console.
tConvertType

Le composant tConvertType permet de convertir automatiquement des données de type Java vers un autre type utilisé par Talend, ce qui permet d'éviter les erreurs de compilation.

Le composant tConvertType permet d’effectuer des conversions spécifiques du type de données Java vers un autre type de données Talend au cours de l’exécution.

Propriétés du tConvertType Standard

Ces propriétés sont utilisées pour configurer le tConvertType s’exécutant dans le framework de Jobs Standard.

Le composant tConvertType Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Repository</td>
<td>Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Auto Cast</td>
<td>Cette case est cochée par défaut. Elle permet de convertir automatiquement des données de type Java.</td>
</tr>
</tbody>
</table>
### Manual Cast

Ce mode n'est pas visible si la case **Auto Cast** est cochée. Elle permet de préciser manuellement les colonnes où une conversion de type Java est nécessaire.

### Set empty values to Null before converting

Cochez cette case pour définir les valeurs des éléments de type **String** ou **Object** comme null pour les données d'entrée.

### Die on error

**Remarque :**
Non disponible pour les Jobs Map/Reduce.

Cette case est cochée pour arrêter le Job lorsque survient une erreur.

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

### Global Variables

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaine de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.
- **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant ne peut être utilisé comme composant de début. Il requiert un flux d’entrée pour fonctionner.</th>
</tr>
</thead>
</table>

### Scénario : Convertir des types de données Java

Le scénario Java suivant permet de créer un Job à quatre composants dans lequel le **tConvertType** est utilisé pour convertir des données de type Java stockées dans trois colonnes différentes et le **tMap** est utilisé pour modifier le schéma : la première colonne reste intacte et les deux dernières colonnes du schéma d’entrée sont fusionnées en une seule colonne dans le schéma de sortie.
Remarque :
Dans ce scénario, le schéma du fichier délimité utilisé en entrée est stocké dans le Repository, vous pouvez donc renseigner les propriétés du composant tFileInputDelimited d’un simple glisser-déposer du répertoire Repository > Metadata > File delimited vers l'espace de modélisation. Pour plus d’informations, consultez le Guide utilisateur du Studio Talend.

Déposer les composants

Procédure
1. Cliquez et déposez les composants suivants dans l’espace de modélisation : tConvertType, tMap et tLogRow.
   La boîte de dialogue [Components] apparaît.
3. Dans la boîte de dialogue, sélectionnez le tFileInputDelimited dans la liste des composants et cliquez sur Ok.
   Un composant tFileInputComponent appelé Java types apparaît dans l’espace de modélisation.

Configurer les composants

Procédure
1. Dans l’espace de modélisation, sélectionnez le composant tFileInputDelimited et cliquez sur la vue Component pour paramétrer ses propriétés de base.
2. Dans l’onglet Basic settings, sélectionnez l’option Repository dans la liste Property Type puisque les propriétés du fichier sont stockées dans le Repository. Les autres champs sont alors renseignés automatiquement grâce aux données récupérées.

Le fichier d’entrée utilisé pour ce scénario est un fichier texte appelé input contenant des données Java de type string (chaîne de caractères), integer (entier) et float.
Renseignez les autres champs, si nécessaire. Pour plus d’informations, consultez `tFileInputDelimited` à la page 1067. Dans ce scénario, il n’y a ni en-tête ni pied de page et il n’y a pas de nombre limite de lignes à traiter.

3. Cliquez sur **Edit schema** pour décrire la structure des données du fichier d’entrée. Dans ce scénario, le schéma est composé de trois colonnes : `StringtoInteger`, `IntegerField` et `FloatToInteger`.

4. Cliquez sur **OK** pour fermer la boîte de dialogue.

5. Dans l’espace de modélisation, sélectionnez le `tConvertType` et cliquez sur la vue **Component** pour paramétrer ses propriétés de base.

6. Dans la liste **Schema Type**, sélectionnez l’option **Built in** et cliquez sur le bouton **Sync columns** pour récupérer automatiquement les colonnes du composant `tFileInputDelimited`.

7. Si nécessaire, cliquez sur **Edit schema** pour décrire manuellement la structure des données de ce composant de traitement.
Dans ce scénario, l'objectif est de convertir les données de type string (chaîne de caractères) en integer (entier) et des données de type float en integer.

Cliquez sur OK pour fermer la boîte de dialogue [Schema of tConvertType].

8. Dans l'espace de modélisation, double-cliquez sur le tMap pour ouvrir l'éditeur du tMap.

L'éditeur du tMap s'ouvre et affiche les noms des colonnes du fichier d'entrée.

9. Dans l'onglet Schema editor en bas de l'éditeur du tMap, cliquez sur le bouton [+] de la table de sortie pour ajouter deux lignes et nommez-les StringToInteger et Sum.

10. Dans l'éditeur du tMap, glissez la ligne StringToInteger de la table d'entrée vers la ligne StringToInteger de la table de sortie.

11. Dans l'éditeur du tMap, glissez les lignes IntegerField et FloatToInteger de la table d'entrée vers la ligne Sum de la table de sortie. Cliquez sur OK pour fermer l'éditeur du tMap.
12. Dans l'espace de modélisation, sélectionnez le *tLogRow* et cliquez sur la vue *Component* pour paramétrer ses propriétés de base. Pour plus d'informations, consultez *tLogRow* à la page 2105.

**Exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour enregistrer votre Job.
2. Appuyez sur **F6** pour l'exécuter.

**Résultats**

```
Starting job tConvertType at 19.03 19/11/2008. 
--------------------------
<p>| tLogRow_1                |</p>
<table>
<thead>
<tr>
<th>StringToInteger</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1579</td>
</tr>
</tbody>
</table>

Job tConvertType ended at 19.03 19/11/2008. [eu]
```

Les données de type string sont converties en integer et affichées dans la colonne *StringToIntegrer* dans la console. Les données de type float sont converties en integer et ajoutées à la valeur *IntegerField* et leur somme est affichée dans la colonne *Sum* dans la console.
tCosmosDBBulkLoad

Ce composant importe des fichiers de données dans différents formats (CSV, TSV ou JSON) dans la base de données Cosmos spécifiée, afin que les données puissent être traitées.

Propriétés du tCosmosDBBulkLoad Standard

Ces propriétés sont utilisées pour configurer le tCosmosDBBulkLoad s’exécutant dans le framework de Jobs Standard.

Le composant tCosmosDBBulkLoad Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|---|---|
| • View schema : sélectionnez cette option afin de voir le schéma.  
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<p>| MongoDB directory | Saisissez dans ce champ le répertoire d’installation de MongoDB. |
| Use replica set address or multiple query routers | Cochez cette case pour afficher la table Server addresses. Dans la table Server addresses, définissez les bases de données MongoDB shardées ou les ensembles de répliques MongoDB auxquel(le)s vous souhaitez vous connecter. |
| Server et Port | Saisissez l’adresse IP et le numéro du port d’écoute du serveur de la base de données. |</p>
<table>
<thead>
<tr>
<th><strong>CosmosDBBulkLoad</strong></th>
<th>Disponible lorsque la case <strong>Use replica set address or multiple query routers</strong> n'est pas cochée.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez le nom de la base de données MongoDB à laquelle vous connecter.</td>
</tr>
<tr>
<td><strong>Collection</strong></td>
<td>Saisissez le nom de la collection dans laquelle importer les données.</td>
</tr>
<tr>
<td><strong>Drop collection if exist</strong></td>
<td>Cochez cette case afin de supprimer la collection si elle existe déjà.</td>
</tr>
<tr>
<td><strong>Authentication mechanism</strong></td>
<td>Parmi les mécanismes listés dans la liste déroulante <strong>Authentication mechanism</strong>, le mécanisme <strong>NEGOTIATE</strong> est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez. Pour plus d’informations sur les autres mécanismes de la liste, consultez <a href="https://docs.mongodb.com/manual/reference/mongo-kerberos/">MongoDB Authentication</a> dans la documentation MongoDB.</td>
</tr>
<tr>
<td><strong>Set Authentication database</strong></td>
<td>Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ <strong>Authentication database</strong> qui s’affiche. Pour plus d’informations sur la base de données d’authentification MongoDB, consultez <a href="https://docs.mongodb.com/manual/reference/mongo-kerberos/">User Authentication database</a> (en anglais).</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez les informations d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres. Ces champs sont disponibles lorsque la case <strong>Use authentication</strong> est cochée. Si le système de sécurité sélectionné dans la liste <strong>Authentication mechanism</strong> est <strong>Kerberos</strong>, saisissez les informations dans les champs suivants <strong>User principal</strong>, <strong>Realm</strong> et <strong>KDC server</strong> et non dans les champs <strong>Username</strong> et <strong>Password</strong>.</td>
</tr>
<tr>
<td><strong>Data file</strong></td>
<td>Saisissez le chemin d’accès complet au fichier duquel importer les données, ou cliquez sur le bouton [...] afin de parcourir votre système jusqu’au fichier de données. Assurez-vous que le fichier de données est bien dans un format standard. Par exemple, les champs des fichiers CSV doivent être séparés par une virgule.</td>
</tr>
<tr>
<td><strong>File type</strong></td>
<td>Sélectionnez le type de fichier dans la liste. Les formats CSV, TSV et JSON sont supportés.</td>
</tr>
<tr>
<td><strong>The JSON file starts with an array</strong></td>
<td>Cochez cette case pour permettre au tCosmosDBBulkLoad de lire les fichiers JSON commençant par un tableau. Cette case est disponible lorsque vous avez sélectionné JSON dans la liste <strong>File type</strong>.</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Action on data</strong></td>
<td>Sélectionnez l’action à effectuer sur les données.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Insert</strong> : insérer des données dans la base de données.</td>
</tr>
<tr>
<td></td>
<td>Lorsque vous insérez des données depuis des fichiers CSV ou TSV dans la base de données MongoDB, vous devez spécifier les champs, en cochant la case <strong>First line is header</strong> ou en les définissant dans le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Upsert</strong> : insérer les données si elles n’existent pas ou les mettre à jour si elles existent.</td>
</tr>
<tr>
<td></td>
<td>Lors d’une opération d’upsert de données dans la base de données MongoDB, vous devez spécifier une liste de champs, pour la partie requête de l’opération.</td>
</tr>
<tr>
<td><strong>Upsert fields</strong></td>
<td>Personnalisez les champs sur lesquels effectuer un upsert.</td>
</tr>
<tr>
<td></td>
<td>Cette table est disponible lorsque vous sélectionnez <strong>Upsert</strong> dans la liste <strong>Action on data</strong>.</td>
</tr>
<tr>
<td><strong>First line is header</strong></td>
<td>Cochez cette case pour utiliser la première ligne d’un fichier CSV ou TSV en tant qu’en-tête.</td>
</tr>
<tr>
<td></td>
<td>Cette case est disponible uniquement lorsque vous sélectionnez CSV ou TSV dans la liste <strong>File type</strong>.</td>
</tr>
<tr>
<td><strong>Ignore blanks</strong></td>
<td>Cochez cette case pour ignorer les champs vides dans les fichiers CSV ou TSV.</td>
</tr>
<tr>
<td></td>
<td>Cette case est disponible uniquement lorsque vous sélectionnez CSV ou TSV dans la liste <strong>File type</strong>.</td>
</tr>
<tr>
<td><strong>Print log</strong></td>
<td>Cochez cette case pour afficher les logs.</td>
</tr>
<tr>
<td><strong>Advanced settings</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Additional arguments</strong></td>
<td>Renseignez cette table afin d’utiliser des arguments supplémentaires selon vos besoins.</td>
</tr>
<tr>
<td></td>
<td>Par exemple, vous pouvez utiliser l’argument &quot;--jsonArray&quot; pour accepter l’import de données de différents documents MongoDB dans un seul tableau JSON. Pour plus d’informations concernant les arguments supplémentaires, consultez le site <a href="http://docs.mongodb.org/manual/reference/program/mongoimport/">http://docs.mongodb.org/manual/reference/program/mongoimport/</a> (en anglais) et lisez la description des options.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau des composants.</td>
</tr>
</tbody>
</table>
### Usage

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec un \texttt{tCosmosDBInput} afin de vérifier si les données sont bien importées.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>L’outil client MongoDB doit être installé sur la machine où les Jobs contenant ce composant sont exécutés.</td>
</tr>
</tbody>
</table>
tCosmosDBConnection

Ce composant crée une connexion à une base de données CosmosDB et réutilise cette connexion dans d'autres composants.

Propriétés du CosmosDBConnection Standard

Ces propriétés sont utilisées pour configurer le tCosmosDBConnection s'exécutant dans le framework des Jobs Standard.

Le composant tCosmosDBConnection Standard appartient à la famille Cloud et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Basic settings**

| API | Sélectionnez l’API de la base de données à utiliser. Les paramètres à définir sont affichés dans la vue **Component**. Dans la dernière version de ce composant, seule l’API de MongoDB est supportée. C’est pour cette raison que la base de données MongoDB est souvent mentionnée dans la documentation des composants CosmosDB. |
| Use replica set address or multiple query routers | Cochez cette case pour afficher la table **Server addresses**. Dans la table **Server addresses**, définissez les bases de données MongoDB shardées ou les ensembles de répliques MongoDB auxquel(le)s vous souhaitez vous connecter. |
| Server et Port | Saisissez l’adresse IP et le numéro du port d’écoute du serveur de la base de données. Disponible lorsque la case **Use replica set address or multiple query routers** n’est pas cochée. |
| Database | Saisissez le nom de la base de données MongoDB à laquelle vous connecter. |
| Authentication mechanism | Parmi les mécanismes listés dans la liste déroulante **Authentication mechanism**, le mécanisme **NEGOTIATE** est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez. Pour plus d’informations sur les autres mécanismes de la liste, consultez MongoDB Authentication (en anglais) dans la documentation MongoDB. |
| Set Authentication database | Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette |
case pour saisir le nom de la base de données en question dans le champ **Authentication database** qui s’affiche.

Pour plus d’informations sur la base de données d’authentification MongoDB, consultez [User Authentication database](en anglais).

### Username et Password
Informations d’authentification de l’utilisateur à la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Ces champs sont disponibles lorsque la case **Use authentication** est cochée.

Si le système de sécurité sélectionné dans la liste **Authentication mechanism** est **Kerberos**, saisissez les informations dans les champs suivants **User principal**, **Realm** et **KDC server** et non dans les champs **Username** et **Password**.

### Advanced settings

#### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

#### No query timeout
Cochez cette case pour empêcher les serveurs MongoDB de fermer les curseurs inactifs après 10 minutes d’inactivité de ces curseurs. Dans cette situation, un curseur inactif reste ouvert jusqu’à ce que les résultats de ce curseur soient épuisés ou que vous fermiez manuellement le curseur à l’aide de la méthode `cursor.close()`.

Un curseur, dans MongoDB, est un pointeur vers l’ensemble de résultats de la requête. Par défaut, c’est-à-dire lorsque la case est décochée, un serveur MongoDB ferme automatiquement les curseurs inactifs après une période d’inactivité donnée, pour éviter les utilisations excessives de mémoire. Pour plus d’informations concernant les curseurs MongoDB, consultez [https://docs.mongodb.org/manual/core/cursors/](https://docs.mongodb.org/manual/core/cursors/) (en anglais).

### Usage

Ce composant est généralement utilisé avec d’autres composants CosmosDB, notamment le **tCosmosClose**.
**tCosmosDBInput**

Ce composant récupère des documents d’une collection dans la base de données Cosmos en fournissant un document de requête contenant les champs auxquels doivent correspondre les documents souhaités.

**Propriétés du tCosmosDBInput Standard**

Ces propriétés sont utilisées pour configurer le tCosmosDBInput s’exécutant dans le framework de Jobs Standard.

Le composant tCosmosDBInput Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>API</strong></td>
<td>Sélectionnez l’API de la base de données à utiliser. Les paramètres à définir sont affichés dans la vue Component.</td>
</tr>
<tr>
<td></td>
<td>Dans la dernière version de ce composant, seule l’API de MongoDB est supportée. C’est pour cette raison que la base de données MongoDB est souvent mentionnée dans la documentation des composants CosmosDB.</td>
</tr>
<tr>
<td><strong>Use replica set address or multiple query routers</strong></td>
<td>Cochez cette case pour afficher la table Server addresses.</td>
</tr>
<tr>
<td></td>
<td>Dans la table Server addresses, définissez les bases de données MongoDB shardées ou les ensembles de répliques MongoDB auxquel(le)s vous souhaitez vous connecter.</td>
</tr>
<tr>
<td><strong>Server et Port</strong></td>
<td>Saisissez l’adresse IP et le numéro du port d’écoute du serveur de la base de données. Disponible lorsque la case Use replica set address or multiple query routers n’est pas cochée.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez le nom de la base de données MongoDB à laquelle vous connecter.</td>
</tr>
<tr>
<td><strong>Set read preference</strong></td>
<td>Cochez cette case et, dans la liste Read preference qui s’affiche, sélectionnez le membre auquel vous souhaitez adresser les opérations de lecture.</td>
</tr>
<tr>
<td></td>
<td>Si vous laissez décochée cette case, le Job utilise les préférences de lecture par défaut. Autrement dit, il utilise le membre primaire d’un Replica set.</td>
</tr>
</tbody>
</table>
Pour plus d'informations, consultez la documentation de MongoDB relative à la réplication et à ses préférences de lecture (Read preferences).

### Authentication mechanism

Parmi les mécanismes listés dans la liste déroulante **Authentication mechanism**, le mécanisme **NEGOTIATE** est recommandé si vous n'utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d'authentification le plus adapté à la version de MongoDB que vous utilisez.

Pour plus d'informations sur les autres mécanismes de la liste, consultez [MongoDB Authentication](https://docs.mongodb.com/manual/reference/permissions/) (en anglais) dans la documentation MongoDB.

### Set Authentication database

Si le nom d'utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d'authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ **Authentication database** qui s'affiche.

Pour plus d'informations sur la base de données d'authentification MongoDB, consultez [User Authentication database](https://docs.mongodb.com/manual/reference/permissions/) (en anglais).

### Username et Password

Informations d'authentification de l'utilisateur à la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Ces champs sont disponibles lorsque la case **Use authentication** est cochée.

Si le système de sécurité sélectionné dans la liste **Authentication mechanism** est Kerberos, saisissez les informations dans les champs suivants **User principal**, **Realm** et **KDC server** et non dans les champs **Username** et **Password**.

### Collection

Nom de la collection dans la base de données.

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Si une colonne dans une base de données est un document JSON et que vous devez lire le document entier, saisissez un astérisque "*" dans la colonne DB column, sans guillemet.

### Query

Spécifiez la condition de la requête. Ce champ est disponible uniquement lorsque vous avez sélectionné Find query dans la liste Query type.

Par exemple, saisissez "{id:4}" afin de récupérer l’enregistrement dont l’id est 4, dans la collection spéciﬁée dans le champ Collection.

À la différence des instructions de requête requises dans le logiciel client MongoDB, la requête fait ici référence au contenu dans `find()`. Ainsi la requête est `{id:4}` alors qu’elle devrait être `db.blog.find({id:4})` dans le client MongoDB.

### Mapping

Chaque colonne du schéma défini pour ce composant représente un champ des documents à lire. Dans cette table, vous devez spécifier les nœuds parents de ces champs, s’il y en a.

Par exemple, dans le document se présentant comme suit :

```json
{
  _id: ObjectId("5099803df4948bd2f98391"),
  person: { first: "Joe", last: "Walker" }
}
```

Les champs first et last ont un nœud père person mais le champ _id ne contient aucun nœud père. Cela fait, la table Mapping doit ressembler à ceci :

<table>
<thead>
<tr>
<th>Column</th>
<th>Parent node path</th>
</tr>
</thead>
<tbody>
<tr>
<td>_id</td>
<td></td>
</tr>
<tr>
<td>first</td>
<td>&quot;person&quot;</td>
</tr>
<tr>
<td>last</td>
<td>&quot;person&quot;</td>
</tr>
</tbody>
</table>

### Sort by

Spécifiez la colonne et choisissez l’ordre pour l’opération de tri.

Ce champ est disponible uniquement lorsque vous avez sélectionné Find query dans la liste Query type.

### Limit

Saisissez le nombre maximal d’enregistrements à récupérer.

Ce champ est disponible uniquement lorsque vous avez sélectionné Find query dans la liste Query type.
### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>
| No query timeout         | Cochez cette case pour empêcher les serveurs MongoDB de fermer les curseurs inactifs après 10 minutes d’inactivité de ces curseurs. Dans cette situation, un curseur inactif reste ouvert jusqu’à ce que les résultats de ce curseur soient épuisés ou que vous fermetez manuellement le curseur à l’aide de la méthode `cursor.close()`.

Un curseur, dans MongoDB, est un pointeur vers l’ensemble de résultats de la requête. Par défaut, c’est-à-dire lorsque la case est décochée, un serveur MongoDB ferme automatiquement les curseurs inactifs après une période d’inactivité donnée, pour éviter les utilisations excessives de mémoire. Pour plus d’informations concernant les curseurs MongoDB, consultez [https://docs.mongodb.org/manual/core/cursors/](https://docs.mongodb.org/manual/core/cursors/) (en anglais). |
| Enable external sort     | Puisque les étapes des pipelines d’agrégation ont une limite d’utilisation de mémoire maximum (100 mégaoctets) et qu’une étape dépassant cette limite produit des erreurs, lors du traitement de jeux de données volumineux, cochez cette case pour éviter que les étapes d’agrégation dépassent cette limite.

Pour plus d’informations concernant ce tri externe, consultez [Large sort operation with external sort](https://docs.mongodb.org/manual/core/cursors/) (en anglais). |

### Usage

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Le composant <strong>tCosmosDBInput</strong> est un composant de début et vous permet de récupérer des enregistrements d’une collection, dans la base de données Cosmos et de les transférer au composant suivant, pour affichage ou stockage.</th>
</tr>
</thead>
</table>
**tCosmosDBOutput**

Ce composant insère, met à jour, insère et met à jour ou supprime des documents dans une collection d’une base de données Cosmos, à partir du flux entrant du composant précédent dans le Job.

**Propriétés du tCosmosDBOutput Standard**

Ces propriétés sont utilisées pour configurer le tCosmosDBOutput s’exécutant dans le framework de Jobs Standard.

Le composant tCosmosDBOutput Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>API</td>
<td>Sélectionnez l’API de la base de données à utiliser. Les paramètres à définir sont affichés dans la vue <strong>Component</strong>. Dans la dernière version de ce composant, seule l’API de MongoDB est supportée. C’est pour cette raison que la base de données MongoDB est souvent mentionnée dans la documentation des composants CosmosDB.</td>
</tr>
<tr>
<td>Use replica set address or multiple query routers</td>
<td>Cochez cette case pour afficher la table <strong>Server addresses</strong>. Dans la table <strong>Server addresses</strong>, définissez les bases de données MongoDB shardées ou les ensembles de répliques MongoDB auxquel(les) vous souhaitez vous connecter.</td>
</tr>
<tr>
<td>Server et Port</td>
<td>Saisissez l’adresse IP et le numéro du port d’écoute du serveur de la base de données. Disponible lorsque la case <strong>Use replica set address or multiple query routers</strong> n’est pas cochée.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données MongoDB à laquelle vous connecter.</td>
</tr>
</tbody>
</table>
### Bulk write

Cochez cette case pour insérer, mettre à jour ou supprimer des données en masse. Notez que cette fonctionnalité est disponible uniquement lorsque la version de MongoDB que vous utilisez est 2.6 ou supérieure.

Vous devez sélectionner **Ordered** ou **Unordered** afin de définir comment la base de données MongoDB traite les données envoyées par le Studio.

- Si vous sélectionnez **Ordered**, MongoDB traite les données en séquence.
- Si vous sélectionnez **Unordered**, MongoDB optimise les opérations d’écriture en masse sans conserver l’ordre dans lequel les opérations individuelles ont été insérées dans l’écriture en masse.


### Authentication mechanism

Parmi les mécanismes listés dans la liste déroulante **Authentication mechanism**, le mécanisme **NEGOTIATE** est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez.

Pour plus d’informations sur les autres mécanismes de la liste, consultez [MongoDB Authentication](http://docs.mongodb.org/manual/reference/permissions/)(en anglais) dans la documentation MongoDB.

### Set Authentication database

Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ **Authentication database** qui s’affiche.


### Username et Password

Informations d’authentification de l’utilisateur à la base de données.

Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Ces champs sont disponibles lorsque la case **Use authentication** est cochée.

Si le système de sécurité sélectionné dans la liste **Authentication mechanism** est **Kerberos**, saisissez les informations dans les champs suivants **User principal**, **User principal**.
<table>
<thead>
<tr>
<th><strong>Realm et KDC server</strong> et non dans les champs <strong>Username</strong> et <strong>Password</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collection</strong></td>
</tr>
<tr>
<td><strong>Drop collection if exist</strong></td>
</tr>
<tr>
<td><strong>Action on data</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job.

<table>
<thead>
<tr>
<th>Built-In</th>
<th>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <strong>Guide utilisateur du Studio Talend</strong>.</th>
</tr>
</thead>
</table>
| Repository | Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.  
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.  
Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com). |

**Mapping**

Chaque colonne du schéma défini pour ce composant représente un champ des documents à lire. Dans cette table, vous devez spécifier les nœuds parents de ces champs, s’il y en a.

Par exemple, dans le document se présentant comme suit :

```json
{
   _id: ObjectId("5099803df3f4948bd2f98391"),
   person: { first: "Joe", last: "Walker" }
}
```

Les champs `first` et `last` ont un nœud père `person` mais le champ `_id` ne contient aucun nœud père. Cela fait, la table **Mapping** doit ressembler à ceci :

<table>
<thead>
<tr>
<th>Column</th>
<th>Parent node path</th>
</tr>
</thead>
<tbody>
<tr>
<td>_id</td>
<td></td>
</tr>
<tr>
<td>first</td>
<td>&quot;person&quot;</td>
</tr>
<tr>
<td>last</td>
<td>&quot;person&quot;</td>
</tr>
</tbody>
</table>

Ce tableau est indisponible lorsque la case **Generate JSON Document** est cochée dans l’onglet **Advanced settings**.

**Die on error**

Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

**Advanced settings**

| Generate JSON Document | Cochez cette case pour la configuration JSON.  
**Configure JSON Tree** : Cliquez sur le bouton [...] pour ouvrir l’interface de configuration de l’arborescence JSON. Pour plus d’informations, consultez [Configurer une arborescence JSON](#) à la page 4287. |
**Group by** : Cliquez sur le bouton [+ ] afin d’ajouter des lignes et sélectionner les colonnes d’entrée pour grouper les enregistrements.

**Remove root node** : Cochez cette case pour supprimer le nœud racine.

**Data node et Query node** (disponibles pour les actions Update et Upsert) : Saisissez le nom du nœud de données et du nœud de requêtes configurés dans l’arborescence JSON.

Ces nœuds sont obligatoires pour les actions Update et Upsert. Ils permettent d’activer les actions Update et Upsert mais ne seront pas stockés dans la base de données.

**No query timeout**

Cochez cette case pour empêcher les serveurs MongoDB de fermer les curseurs inactifs après 10 minutes d’inactivité de ces curseurs. Dans cette situation, un curseur inactif reste ouvert jusqu’à ce que les résultats de ce curseur soient épuisés ou que vous fermiez manuellement le curseur à l’aide de la méthode `cursor.close()`.

Un curseur, dans MongoDB, est un pointeur vers l’ensemble de résultats de la requête. Par défaut, c’est-à-dire lorsque la case est décochée, un serveur MongoDB ferme automatiquement les curseurs inactifs après une période d’inactivité donnée, pour éviter les utilisations excessives de mémoire. Pour plus d’informations concernant les curseurs MongoDB, consultez [https://docs.mongodb.org/manual/core/cursors/](https://docs.mongodb.org/manual/core/cursors/) (en anglais).

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

---

**Usage**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Le <strong>tCosmosDBOutput</strong> exécute l’action définie sur la collection dans la base de données MongoDB, à partir du flux entrant depuis le composant précédent dans le Job.</th>
</tr>
</thead>
</table>

**Limitation**

- Le paramètre "multi", qui vous permet de mettre à jour plusieurs documents en même temps, n’est pas supporté. Par conséquent, si deux documents ont la même clé, le premier est toujours mis à jour, mais le second ne l’est jamais.
- Pour l’opération de mise à jour, la clé ne peut être un tableau (array) JSON.
**tCosmosDBRow**

Ce composant exécute les commandes et les fonctions de la base de données Cosmos.

**Propriétés du tCosmosDBRow Standard**

Ces propriétés sont utilisées pour configurer le tCosmosDBRow s'exécutant dans le framework de Jobs Standard.

Le composant tCosmosDBRow Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

### Basic settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>API</td>
<td>Sélectionnez l'API de la base de données à utiliser. Les paramètres à définir sont affichés dans la vue Component. Dans la dernière version de ce composant, seule l'API de MongoDB est supportée. C'est pour cette raison que la base de données MongoDB est souvent mentionnée dans la documentation des composants CosmosDB.</td>
</tr>
<tr>
<td>Use replica set address or multiple query routers</td>
<td>Cochez cette case pour afficher la table Server addresses. Dans la table Server addresses, définissez les bases de données MongoDB shardées ou les ensembles de répliques MongoDB auxquel(le)s vous souhaitez vous connecter.</td>
</tr>
<tr>
<td>Server et Port</td>
<td>Saisissez l'adresse IP et le numéro du port d'écoute du serveur de la base de données. Disponible lorsque la case Use replica set address or multiple query routers n’est pas cochée.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données MongoDB à laquelle vous connecter.</td>
</tr>
<tr>
<td>Authentication mechanism</td>
<td>Parmi les mécanismes listés dans la liste déroulante Authentication mechanism, le mécanisme NEGOTIATE est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez. Pour plus d’informations sur les autres mécanismes de la liste, consultez MongoDB Authentication (en anglais) dans la documentation MongoDB.</td>
</tr>
</tbody>
</table>
## Set Authentication database

Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ **Authentication database** qui s’affiche.

Pour plus d’informations sur la base de données d’authentification MongoDB, consultez **User Authentication database** (en anglais).

## Username et Password

Informations d’authentification de l’utilisateur à la base de données.

Pour saisir le mot de passe, cliquez sur le bouton `[...]` à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Ces champs sont disponibles lorsque la case **Use authentication** est cochée.

Si le système de sécurité sélectionné dans la liste **Authentication mechanism** est Kerberos, saisissez les informations dans les champs suivants **User principal**, **Realm** et **KDC server** et non dans les champs **Username** et **Password**.

## Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

## Execute command

Cochez cette case afin de saisir des commandes MongoDB dans le champ **Command** pour l’exécution.

- **Command** : dans ce champ, saisissez la commande à exécuter, si cette commande contient une seule variable.
Par exemple, si vous devez construire la commande suivante :

- 

\{"isMaster": 1\}

Saisissez simplement l'isMaster entre guillements.

- **Construct command from keys and values** : si la commande à exécuter contient différentes variables, cochez cette case et, dans la table **Command keys and values**, ajoutez les variables et leurs valeurs à utiliser.

Par exemple, si vous devez construire la commande suivante :

\{
    renameCollection :
    "<source_namespace>",
    to : "<target_namespace>",
    dropTarget : < true | false >
\}

Vous devez ajouter trois lignes à la table **Command keys and values** et saisir une paire variable-valeur pour chaque ligne, entre guillemets :

- 

"renameCollection"
  "old_name"
  "to"
  "new_name"
  "dropTarget"
  "false"

- **Construct command from a JSON string** : si vous souhaitez saisir directement la commande à utiliser, cochez cette case et saisissez cette commande dans le champ **JSON string command** affiché.

Une commande est autorisée par composant **tCosmosDBRow**.

Par exemple :

\"{createIndexes: 'restaurants',
    indexes : [{key :
        {restaurant_id: 1}, name: 'id_index_2', unique: true}]}\"

Notez que vous devez utiliser des guillemets simples pour entourer les valeurs String utilisées dans la commande et des guillemets doubles pour entourer la commande elle-même.

Pour plus d'informations concernant les commandes MongoDB que vous pouvez utiliser dans ce champ, consultez https://docs.mongodb.org/manual/reference/command/ (en anglais).

**Die on error**

Cette case est décochée par défaut, afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Utilisation

| Règle d’utilisation | Le composant tCosmosDBRow vous permet de manipuler la base de données Cosmos à l’aide des commandes et des fonctions MongoDB. |
**tCouchbaseInput**

Ce composant interroge des documents dans une base de données Couchbase.

### Propriétés du tCouchbaseInput Standard

Ces propriétés sont utilisées pour configurer le tCouchbaseInput s’exécutant dans le framework de Jobs Standard.

Le composant tCouchbaseInput Standard appartient à la famille Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

#### Basic settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>Saisissez, entre guillemets doubles, le nom du bucket de données dans la base de données Couchbase. Si vous utilisez Couchbase V5.0, ce nom de bucket est le nom de l’utilisateur que vous avez créé dans l’onglet Security de votre interface Couchbase.</td>
</tr>
<tr>
<td>Password</td>
<td>Fournissez les identifiants d’authentification à un bucket. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. Si vous utilisez Couchbase V5.0, saisissez la valeur saisie dans le champ Bucket comme mot de passe, car, dans Couchbase V5.0, aucun mot de passe n’est associé à un bucket. Cependant, dans Couchbase, vous devez créer un utilisateur ayant un rôle avec les droits appropriés pour accéder aux buckets. Pour plus d’informations concernant le contrôle des accès et d’autres prérequis importants côté Couchbase V5.0, consultez les Release Notes de Couchbase V5.0 : <em>Major Behavior Changes</em> (en anglais).</td>
</tr>
<tr>
<td>Bootstrap nodes</td>
<td>Saisissez le nom ou l’adresse IP du nœud de données sur lequel Couchbase SDK doit effectuer un bootstrap. Comme Couchbase recommande de spécifier plusieurs nœuds sur lesquels effectuer un bootstrap, saisissez les noms ou les adresses IP de ces nœuds dans le champ, en les séparant à l’aide d’une virgule (,). Pour plus d’informations concernant le bootstrapping Couchbase, consultez <em>How Couchbase SDKs connect to the cluster</em> (en anglais).</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark,</td>
</tr>
</tbody>
</table>
évitez le mot réservé `line` lors du nommage des champs.

Le schéma de ce composant est en lecture seule. La colonne `content` stocke les documents à utiliser, la colonne `key` stocke les IDs de ces documents et les autres colonnes stockent les informations techniques Couchbase.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Variables globales

| Variables globales | `NB_LINE` : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable `After` et retourne un entier.  

`ERROR_MESSAGE` : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case `Die on error` est décochée, si le composant a cette option.  

Une variable `Flow` fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant.  

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Le `tCouchbasInput` est un composant de début et lit des documents de la base de données Couchbase. |
tCouchbaseOutput

Ce composant effectue des actions d’upsert sur des documents dans une base de données Couchbase selon les données plates provenant des composants précédents.

Le tCouchbaseOutput ajoute un nouveau document ou remplace sa valeur s’il existe déjà.

Propriétés du tCouchbaseOutput Standard

Ces propriétés sont utilisées pour configurer le tCouchbaseOutput s’exécutant dans le framework de Jobs Standard.

Le composant tCouchbaseOutput Standard appartient à la famille Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>Saisissez, entre guillemets doubles, le nom du bucket de données dans la base de données Couchbase. Si vous utilisez Couchbase V5.0, ce nom de bucket est le nom de l’utilisateur que vous avez créé dans l’onglet Security de votre interface Couchbase.</td>
</tr>
<tr>
<td>Password</td>
<td>Fournissez les identifiants d’authentification à un bucket. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. Si vous utilisez Couchbase V5.0, saisissez la valeur saisie dans le champ Bucket comme mot de passe, car, dans Couchbase V5.0, aucun mot de passe n’est associé à un bucket. Cependant, dans Couchbase, vous devez créer un utilisateur ayant un rôle avec les droits appropriés pour accéder aux buckets. Pour plus d’informations concernant le contrôle des accès et d’autres prérequis importants côté Couchbase V5.0, consultez les Release Notes de Couchbase V5.0 : Major Behavior Changes (en anglais).</td>
</tr>
<tr>
<td>Bootstrap nodes</td>
<td>Saisissez le nom ou l’adresse IP du nœud de données sur lequel Couchbase SDK doit effectuer un bootstrap. Comme Couchbase recommande de spécifier plusieurs nœuds sur lesquels effectuer un bootstrap, saisissez les noms ou les adresses IP de ces nœuds dans le champ, en les séparant à l’aide d’une virgule (,). Pour plus d’informations concernant le bootstrapping Couchbase, consultez How Couchbase SDKs connect to the cluster (en anglais).</td>
</tr>
</tbody>
</table>
Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur `Edit schema` pour modifier le schéma. Si le schéma est en mode `Repository`, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode `Built-In` et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur `No` et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton `Sync columns` pour récupérer le schéma du composant précédent.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

**Field to use as ID**

Saisissez, entre guillemets doubles, le nom de la colonne du schéma du tCouchbaseOutput afin de fournir les ID pour les documents à écrire dans Couchbase.

**Die on error**

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

Décrochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.
## Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.

---

## Global Variables

**Variables globales**

**NB_LINE**: nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.

**NB_SUCCESS**: nombre de lignes traitées avec succès. Cette variable est une variable *After* et retourne un nombre entier.

**NB_REJECT**: nombre de lignes rejetées. Cette variable est une variable *After* et retourne un nombre entier.

**ERROR_MESSAGE**: message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l'exécution d'un composant. Une variable *After* fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

---

## Utilisation

**Règle d'utilisation**

Précédé par un composant d'entrée, le **tCouchbaseOutput** écrit des données plates dans des documents afin de les stocker dans une base de données Couchbase.
tCouchDBClose

Ce composant ferme une connexion active à un serveur CouchDB, afin de libérer des ressources occupées.

Propriétés du tCouchDBClose Standard

Ces propriétés sont utilisées pour configurer le tCouchDBClose s'exécutant dans le framework de Jobs Standard.

Le composant tCouchDBClose Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Component List | Sélectionnez une connexion CouchDB active à fermer. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants CouchDB, notamment le tCouchDBConnec tion. |
Scénario associé

Pour un scénario utilisant un tCouchDBClose, consultez Scénario : Répliquer des données de la base de données source à la base de données cible à la page 568.
tCouchDBConnection

Ce composant ouvre une connexion à un serveur CouchDB pouvant être réutilisée par d’autres composants CouchDB.

Vous pouvez également configurer les paramètres de réplication si une réplication est déclenchée entre la base de données source et la base de données cible.

**Propriétés du tCouchDBConnection Standard**

Ces propriétés sont utilisées pour configurer le tCouchDBConnection s’exécutant dans le framework de Jobs Standard.

Le composant tCouchDBConnection Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DB Version</strong></td>
<td>Sélectionnez la version de CouchDB que vous utilisez.</td>
</tr>
<tr>
<td><strong>Server</strong></td>
<td>Saisissez l’adresse IP ou le nom de l’hôte du serveur CouchDB.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Saisissez le numéro du port d’écoute du serveur CouchDB.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez le nom de la base de données à utiliser.</td>
</tr>
<tr>
<td><strong>Required authentication</strong></td>
<td>Cochez cette case afin de fournir les informations d’authentification pour CouchDB.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Username</strong> : Saisissez le nom d’utilisateur de CouchDB.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Password</strong> : Saisissez le mot de passe pour CouchDB.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Replicate target database</strong></td>
<td>Cochez cette case afin de configurer la réplication dans la table qui apparaît.</td>
</tr>
<tr>
<td></td>
<td><strong>Target DB name</strong> : Spécifiez la base de données cible dans laquelle les documents seront copiés. Vous pouvez saisir le nom d’une base de données locale ou l’URL d’une base de données distante.</td>
</tr>
<tr>
<td></td>
<td><strong>Continuous</strong> : Cochez cette case pour continuer la réplication après le redémarrage du serveur.</td>
</tr>
<tr>
<td></td>
<td><strong>Create target DB</strong> : Cochez cette case pour créer la base de données cible si elle n’existe pas.</td>
</tr>
</tbody>
</table>
**tCouchDBConnection**

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants CouchDB, notamment le tCouchDBClose. |

**Scénario associé**

Pour un scénario utilisant un tCouchDBConnection, consultez Scénario : Répliquer des données de la base de données source à la base de données cible à la page 568.
tCouchDBInput

Ce composant extrait les données JSON souhaitées d’une base de données CouchDB afin de les transformer, de les migrer vers un format différent ou de les traiter avant de les réinsérer dans la base de données.

**Propriétés du tCouchDBInput Standard**

Ces propriétés sont utilisées pour configurer le tCouchDBInput s’exécutant dans le framework de Jobs Standard.

Le composant tCouchDBInput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

• **View schema** : sélectionnez cette option afin de voir le schéma.

• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Les colonnes du schéma dépendent de votre configuration :

• Si vous cochez les cases Query by view et Is reduce en même temps et que vous spécifiez un niveau de groupe après avoir coché la case Group, seules les colonnes key et value sont disponibles dans le schéma.

• Si vous cochez la case Include docs mais ne cochez pas la case Is reduce, les colonnes id, key, value et JSONDoc sont disponibles dans le schéma. |
- Si les deux cases **Is reduce** et **Include docs** sont décochées, les colonnes **id**, **key** et **value** sont disponibles dans le schéma.

<table>
<thead>
<tr>
<th>Use existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version de CouchDB que vous utilisez.</td>
</tr>
<tr>
<td>Server</td>
<td>Nom de l’hôte ou adresse IP du serveur CouchDB.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur CouchDB.</td>
</tr>
<tr>
<td>Database</td>
<td>Spécifiez la base de données à utiliser.</td>
</tr>
<tr>
<td>Required authentication</td>
<td>Cochez cette case afin de fournir les informations d’authentification pour CouchDB.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Username</strong> : Saisissez le nom d’utilisateur de CouchDB.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Password</strong> : Saisissez le mot de passe pour CouchDB. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Use trigger replication</td>
<td>Cochez cette case pour déclencher la réplication entre les bases de données.</td>
</tr>
</tbody>
</table>
|                         | **Avertissement** :
|                         | Cette case apparaît uniquement lorsque vous ne cochez pas la case **Use an existing connection**. |
| Target DB name          | Spécifiez la base de données cible dans laquelle copier les documents. Spécifiez le nom d’une base de données locale ou l’URL d’une base de données distante. |
| Continuous              | Cochez cette case pour continuer la réplication après le redémarrage du serveur. |
| Create target DB        | Cochez cette case pour créer la base de données cible si elle n’existe pas. |
| IS Canceled             | Cochez cette case pour annuler la réplication entre la base de données source spécifiée et la base de données cible à la fin du Job. |
| Query by view           | Cochez cette case pour spécifier des conditions de requête basées sur une vue contenant une fonction de mapping et une fonction Reduce facultative. |
**Action on design document**

Sélectionnez l’opération à effectuer sur le document Design Document qui vous intéresse :

- **None** : Aucune opération à effectuer.
- **Drop and create design document** : Le Design Document est supprimé et créé à nouveau.
- **Create design document** : Un nouveau Design Document est créé.
- **Create design document if not exists** : Le Design Document est créé s’il n’existe pas.
- **Drop design document if exists and create** : Le Design Document est supprimé s’il existe et recréé.

**View**

Saisissez le nom de la vue à partir de laquelle lire les données.

**Action on view**

Sélectionnez l’opération à effectuer sur la vue souhaitée :

- **None** : Aucune opération à effectuer.
- **Create view** : Une nouvelle vue est créée.

**Map**

Saisissez la fonction map dans ce champ.

⚠️ **Avertissement** :

Ce champ apparaît uniquement lorsque vous sélectionnez **Create view** dans la liste déroulante **Action on view**.

**Reduce**

Saisissez la fonction reduce dans ce champ.

⚠️ **Avertissement** :

Ce champ apparaît uniquement lorsque vous sélectionnez **Create view** dans la liste déroulante **Action on view** et que vous cochez la case **Is reduce**.

**Start key**

Saisissez la clé de début.

**End key**

Saisissez la clé de fin.

**Start key docid**

Saisissez l’ID du document de la clé de début.

**End key docid**

Saisissez l’ID du document de la clé de fin.

**Is reduce**

Cochez cette case pour que la fonction reduce prenne effet.

**Group**

Cochez cette case pour réduire la fonction reduce à un ensemble de clés distinctes ou à une seule ligne de résultat.

⚠️ **Avertissement** :

Cette case apparaît lorsque vous cochez la case **Is reduce**.
<table>
<thead>
<tr>
<th><strong>Group level</strong></th>
<th>Saisissez le niveau du groupe spécifique dans ce champ après avoir coché la case <strong>Group</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Include docs</strong></td>
<td>Cochez cette case pour inclure le document ayant émis chaque entrée de vue. <strong>Avertissement :</strong> Cette case apparaît lorsque vous ne cochez pas la case <strong>Is reduce</strong>.</td>
</tr>
<tr>
<td><strong>Descending</strong></td>
<td>Cochez cette case pour inverser l’ordre des requêtes.</td>
</tr>
<tr>
<td><strong>Add options</strong></td>
<td>Cochez cette case pour ajouter plus d’options de requêtes et configurer les paramètres selon vos besoins.</td>
</tr>
<tr>
<td><strong>Extract JSON field</strong></td>
<td>Cochez cette case pour extraire les données JSON souhaitées selon la requête XPath.</td>
</tr>
<tr>
<td><strong>JSON field</strong></td>
<td>Liste des champs JSON à extraire. <strong>Avertissement :</strong> Assurez-vous d’avoir sélectionné un champ JSON dont la valeur est un document JSON. Sinon, des erreurs peuvent survenir.</td>
</tr>
<tr>
<td><strong>Loop XPath query</strong></td>
<td>Nœud dans le champ JSON sur lequel la boucle est basée.</td>
</tr>
<tr>
<td><strong>Mapping</strong></td>
<td><strong>Schema output column</strong> : schéma défini pour contenir les données extraites d’un champ JSON. <strong>XPath query</strong> : saisissez le nom du nœud à extraire du champ JSON. <strong>Get Nodes</strong> : cochez cette case si vous devez récupérer de valeurs d’un nœud dans le champ JSON.</td>
</tr>
<tr>
<td><strong>Limit</strong></td>
<td>Saisissez le nombre maximal de lignes à traiter.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

**Global Variables**

| **Global Variables** | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. |
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | Ce composant nécessite un lien de sortie. |

### Scénario associé

Pour un scénario utilisant un **tCouchDBInput**, consultez **Scénario : Répliquer des données de la base de données source à la base de données cible** à la page 568.
tCouchDBOutput

Le tCouchDBOutput vous permet de charger des documents JSON, d’y écrire ou supprimer des données et de sauvegarder ces documents dans la base de données d’un serveur CouchDB.

Le composant tCouchDBOutput reçoit des données du composant précédent et les écrit dans CouchDB.

Propriétés du tCouchDBOutput Standard

Ces propriétés sont utilisées pour configurer le tCouchDBOutput s’exécutant dans le framework de Jobs Standard.

Le composant tCouchDBOutput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot reservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
| | • View schema : sélectionnez cette option afin de voir le schéma.
| | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
| | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |

| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets. |
Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

<table>
<thead>
<tr>
<th>Sync columns</th>
<th>Cliquez sur ce bouton pour récupérer le schéma du composant précédent dans le Job.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version de CouchDB que vous utilisez.</td>
</tr>
<tr>
<td>Server</td>
<td>Nom de l’hôte ou adresse IP du serveur CouchDB.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute du serveur CouchDB.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données à utiliser.</td>
</tr>
</tbody>
</table>
| Required authentication | Cochez cette case afin de fournir les informations d’authentification pour CouchDB.  
  - **Username** : Saisissez le nom d’utilisateur de CouchDB.  
  - **Password** : Saisissez le mot de passe pour CouchDB.  
  Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Use trigger replication | Cochez cette case pour déclencher la réplication entre les bases de données.  
  **Avertissement** :  
  Cette case apparaît uniquement lorsque vous ne cochez pas la case **Use an existing connection**. |
| Target DB name | Spécifiez la base de données cible dans laquelle copier les documents. Vous pouvez saisir le nom d’une base de données locale ou l’URL d’une base de données distante. |
| Continuous | Cochez cette case pour continuer la réplication après le redémarrage du serveur. |
| Create target DB | Cochez cette case pour créer la base de données cible si elle n’existe pas. |
| ISCanceled | Cochez cette case pour annuler la réplication entre la base de données source spécifiée et la base de données cible à la fin du Job. |
| Action on data | Sélectionnez l’opération à effectuer sur les données qui vous intéressent : |
| **tCouchDBOutput** | • **Insert** : Insérer des données dans la base de données.  
  • **Update** : Mettre à jour les données dans la base de données.  
  • **Upsert** : Insérer les données si elles n’existent pas ou mettre à jour les données existantes.  
  • **Delete** : Supprimer les données de la base de données. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Generate JSON Document</strong></td>
<td>Cochez cette case pour générer un document JSON et configurer la structure de données que vous souhaitez utiliser.</td>
</tr>
<tr>
<td><strong>Key</strong></td>
<td>Sélectionnez la clé à utiliser, dans la liste déroulante.</td>
</tr>
<tr>
<td><strong>Configure JSON Tree</strong></td>
<td>Cliquez sur le bouton […] pour ouvrir la fenêtre de configuration de l’arborescence JSON. Pour plus d’informations, consultez Configurer une arborescence JSON à la page 4287.</td>
</tr>
<tr>
<td><strong>Group by</strong></td>
<td>Personnalisez les colonnes d’entrée sur lesquelles vous souhaitez regrouper les données.</td>
</tr>
<tr>
<td><strong>Remove root node</strong></td>
<td>Cochez cette case pour supprimer le nœud racine.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

**Global Variables**

| **Global Variables** |  
  | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.  
  | **NB_LINE INSERTED** : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.  
  | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
  | Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
  | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant est utilisé en tant que composant de sortie et nécessite un lien d’entrée.</td>
<td></td>
</tr>
</tbody>
</table>

### Scénario : Répliquer des données de la base de données source à la base de données cible

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario décrit un Job écrivant des informations concernant des livres dans une base de données CouchDB, répliquant les données dans la base de données cible et affichant certaines informations intéressantes répliquées dans la console.

![Diagramme de composition](image)

### Relier les composants

1. Dépøsez les composants suivants de la Palette dans l’espace de modélisation graphique : un tCouchDBConnection, un tFileInputDelimited, un tCouchDBOutput, un tCouchDBInput, un tLogRow et un tCouchDBClose.
2. Reliez le tFileInputDelimited au tCouchDBOutput à l'aide d'un lien Row > Main.
3. De la même façon, reliez le tCouchDBInput au tLogRow.
4. Connectez le tCouchDBConnection au tFileInputDelimited à l'aide d'un lien Trigger > OnSubjobOk.
5. Répétez l'opération pour relier le tFileInputDelimited au tCouchDBInput et le tCouchDBInput au tCouchDBClose.
6. Renommez les composants afin de mieux identifier leur rôle au sein du Job.

Configurer les composants

Ouvrir une connexion CouchDB et déclencher la réplication

Procédure

1. Double-cliquez sur le tCouchDBConnection pour ouvrir sa vue Basic settings.

   ![Open CouchDB Connection](image)

2. Dans les champs Server et Port, saisissez les informations de connexion.
3. Dans le champ Database, saisissez le nom de la base de données à utiliser : bookstore_old dans cet exemple.
4. Sélectionnez la version de CouchDB que vous utilisez, dans la liste DB Version.
5. Cochez la case Use trigger replication.
6. Dans la zone Replicate target database, cliquez sur le bouton [+] pour ajouter une ligne pour les paramètres de réplication.
7. Saisissez le nom de la base de données cible dans la colonne Target DB name : bookstore_new dans cet exemple.
8. Cochez la case Continuous pour continuer la réplication après redémarrage du serveur.
9. Dans cet exemple, la base de données cible n’existe pas. Cochez la case Create target DB pour créer la base de données cible.

Lire les données d’entrée

Procédure

1. Double-cliquez sur le composant tFileInputDelimited pour ouvrir sa vue Component.

<table>
<thead>
<tr>
<th>_id</th>
<th>title</th>
<th>author</th>
<th>category</th>
<th>ISBN</th>
<th>abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Computer Networks: A Systems Approach; Larry L. Peterson, Bruce S. Davie; Computer Science; 0123850606; This best-selling and classic book teaches you the key principles of computer networks with examples drawn from the real world of network and protocol design.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>David Copperfield; Charles Dickens; Language &amp; Literature; 1555763227; This adaptation of the original story is presented in the format of a novel study, complete with exercises and vocabulary lists, and is geared to the language arts classes of grades 4 and 5.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>Life of Pi; Yann Martel; Language &amp; Literature; 0547350651; The son of a zookeeper, Pi Patel has an encyclopedic knowledge of animal behavior and a fervent love of stories.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>Les Miserables: Easyread Comfort Edition; Victor Hugo; Language &amp; Literature; 1425048250; Expressing the author's ideas about society, religion and politics, it is in the backdrop of Napoleonic Wars and ensuing years that the story unravels. Grace, moral philosophy, law and history of France are discussed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>Computer Security; Dieter Gollmann; Computer Science; 0470741155; This text moves away from the 'multi-level' security approach to compare and evaluate design alternatives in computer security.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>006</td>
<td>Advanced Database Systems; Carlo Zaniolo; Database; 155860443X; This book, written by a team of leading specialists in their fields, introduces the research issues at the forefront of database technology and supports them with a variety of examples.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Cliquez sur **Edit schema** pour définir les données à passer au composant **tCouchDBOutput**.

![Schema of Read_info](image)

**Ecrire des données dans une base de données et les répliquer**

**Procédure**

1. Double-cliquez sur le composant **tCouchDBOutput** pour ouvrir sa vue **Component**.

![Write_into_database_and_replicate_the_data (tCouchDBOutput_1)](image)

2. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
3. Cochez la case **Use an existing connection**. Dans cet exemple, la réplication est déclenchée à l’ouverture de la connexion à CouchDB.
4. Sélectionnez **Upsert** dans la liste déroulante **Action on data**.

**Extraire les données répliquées qui vous intéressent**

**Procédure**

1. Double-cliquez sur le composant **tCouchDBInput** afin d’ouvrir sa vue **Component**.
2. Cliquez sur **Edit schema** pour définir la structure des données à lire dans la base de données CouchDB.

Par défaut, la case **Include docs** est cochée, les colonnes **id**, **key**, **value** et **jsonDoc** sont disponibles dans le schéma.

Dans cet exemple, quatre colonnes à extraire sont définies : **id**, **title**, **author** et **category**.

4. Dans le champ Database, saisissez le nom de la base de données de laquelle les données répliquées seront lues. Dans cet exemple, saisissez bookstore_new.

5. Dans la zone Querying options, saisissez la clé de début dans le champ Start key et la clé de fin dans le champ End key, afin de configurer l’écart des données à lire : "001" et "006" dans cet exemple.

6. Cochez la case Extract JSON field pour extraire les données souhaitées.

7. Sélectionnez jsonDoc dans la liste JSON field.

8. Dans la zone Mapping, cliquez sur le bouton [+ ] pour ajouter des éléments. Sélectionnez la colonne du schéma de sortie dans la liste puis saisissez la requête XPath.

Afficher les données extraites

Procédure

1. Double-cliquez sur le composant tLogRow pour ouvrir sa vue Component.

2. Cliquez sur Edit schema pour définir la structure des données à afficher dans la console. Dans cet exemple, vous devez supprimer la colonne jsonDoc.

3. Dans la zone Mode, sélectionnez Table (print values in cells of a table).

Fermer la connexion CouchDB

Procédure

1. Double-cliquez sur le composant tCouchDBClose pour ouvrir sa vue Component.
2. Sélectionnez, dans la liste **Component List**, la connexion à fermer.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.
2. Exécutez le Job en appuyant sur **F6** ou en cliquant sur le bouton **Run** dans la vue Run.

Les informations concernant les livres sont lues depuis la base de données répliquée et affichées dans la console.
**tCreateTable**

Ce composant crée une table pour un type de base de données spécifique.

**Propriétés du tCreateTable Standard**

Ces propriétés sont utilisées pour configurer le tCreateTable s'exécutant dans le framework de Jobs Standard.

Le composant tCreateTable Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Database Type</th>
<th>Sélectionnez le type de la base de données. Les propriétés de connexion peuvent être légèrement différentes selon le type de base de données sélectionné.</th>
</tr>
</thead>
</table>
| Property Type | Sélectionnez la manière de configurer les informations de connexion.  
• **Built-In** : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.  
• **Repository** : les informations de connexion stockées centralement dans le **Repository > Metadata** seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue **Repository Content**, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. |
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
Notez que lorsqu’un Job contient un Job parent et un Job enfant, si vous devez partager une connexion existante entre ces deux niveaux, par exemple pour partager la connexion créée par le Job père au Job fils, vous devez :  
1. au niveau du Job père, enregistrer la connexion à la base de données à partager dans la vue **Basic settings** du composant de connexion créant cette connexion à la base de données.  
2. au niveau du Job fils, utiliser un composant de connexion dédié afin de lire cette connexion enregistrée.  
Pour un exemple concernant le partage de connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**. |
<table>
<thead>
<tr>
<th>DB Version</th>
<th>Sélectionnez la version de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Saisissez l'adresse IP ou le nom d'hôte de la base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d'écoute de la base de données.</td>
</tr>
<tr>
<td>Database name</td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td>Access File</td>
<td>Chemin d'accès au fichier de la base de données Access.</td>
</tr>
<tr>
<td>Firebird File</td>
<td>Chemin d'accès au fichier de la base de données Firebird.</td>
</tr>
<tr>
<td>Interbase File</td>
<td>Chemin d'accès au fichier de la base de données Interbase.</td>
</tr>
<tr>
<td>SQLite File</td>
<td>Chemin d'accès au fichier de la base de données SQLite.</td>
</tr>
<tr>
<td>Running Mode</td>
<td>Sélectionnez le mode d'exécution correspondant à la configuration de votre base de données. Cette propriété est disponible uniquement pour le type de base de données HSQLDb.</td>
</tr>
<tr>
<td>Use TLS/SSL Sockets</td>
<td>Cochez cette case pour activer le mode de sécurité, si nécessaire. Cette propriété est disponible uniquement pour le type de base de données HSQLDb.</td>
</tr>
<tr>
<td>DB Alias</td>
<td>Nom de la base de données. Cette propriété est disponible uniquement pour le type de base de données HSQLDb.</td>
</tr>
<tr>
<td>Framework Type</td>
<td>Sélectionnez le type de framework pour votre base de données. Cette propriété est disponible uniquement pour le type de base de données JavaDb.</td>
</tr>
<tr>
<td>DB Root Path</td>
<td>Parcourez jusqu'à la racine de votre base de données. Cette propriété est disponible uniquement pour le type de base de données JavaDb.</td>
</tr>
<tr>
<td>ODBC Name</td>
<td>Nom de la base de données ODBC.</td>
</tr>
<tr>
<td>Connection Type</td>
<td>Sélectionnez le type de connexion à la base de données Oracle.</td>
</tr>
</tbody>
</table>
- **Oracle SID** : sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.
- **Oracle Service** : sélectionnez ce type de connexion pour utiliser l’alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante.
- **Oracle OCI** : sélectionnez ce type de connexion pour utiliser l’interface d’appel de la base de données Oracle (Oracle Call Interface) accompagnée d’un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle.
- **Oracle Custom** : sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.
- **WALLET** : sélectionnez ce type de connexion pour stocker les informations d’authentification dans un portefeuille (wallet) Oracle.

### Username et Password
Saisissez les données d’authentification de l’utilisateur de la base de données.
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

### Table name
Saisissez le nom de la table à créer.

### Table Action
Sélectionnez l’action à effectuer sur la table.
- **Create table** : la table spécifiée n’existe pas et est créée.
- **Create table if not exists** : la table spécifiée est créée si elle n’existe pas.
- **Drop table if exits and create** : la table est supprimée si elle existe déjà et est créée à nouveau.

### Case Sensitive
Cochez cette case pour rendre sensible à la casse le nom de la colonne/de la table.
Cette propriété est disponible uniquement pour le type de base de données HSQLDb.

### Temporary Table
Cochez cette case si vous souhaitez sauvegarder temporairement la table créée.
Cette propriété est disponible uniquement pour le type de base de données MySQL.

### Create
Sélectionnez le type de table à créer.
- **SET TABLE** : la table n’autorise pas les lignes en doublon.
- **MULTISET TABLE** : la table autorise les lignes en doublon.
Cette propriété est disponible uniquement pour le type de base de données Teradata.

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Advanced settings

<table>
<thead>
<tr>
<th>Additional JDBC Parameters</th>
<th>Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Cette option n'est pas disponible si vous avez coché la case <strong>Use an existing connection</strong> dans la vue <strong>Basic settings</strong>. Cette propriété est disponible uniquement pour les types de bases de données AS/400 et MSSQL Server.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create projection</td>
<td>Cochez cette case pour créer une projection. Cette propriété est disponible uniquement pour le type de base de donnée Vertica.</td>
</tr>
</tbody>
</table>
Global Variables

| QUERY | Instruction de requête en cours de traitement. Cette variable est une variable Flow et retourne une chaîne de caractères. |
| ERROR_MESSAGE | Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job. |

Scénario : Créer une nouvelle table dans une base de données MySQL

Le Job décrit ci-dessous a été conçu pour créer une nouvelle table dans une base de données, composée d'un schéma pris à partir du schéma d'un fichier délimité stocké dans le Repository. Ce Job est formé d'un seul composant.

![tCreateTable_1](image)

Procédure

1. Cliquez et déposez un composant tCreateTable de la famille Databases de la Palette dans l'espace de modélisation graphique.
2. Dans la liste Database type de l'onglet Basic settings de la vue Component, sélectionnez Mysql.

![tCreateTable_1](image)

3. Dans la liste déroulante Table Action, sélectionnez Create table.
4. Ne cochez la case Use Existing Connection uniquement si vous utilisez un composant avec une connexion dédiée (pour plus d’informations, consultez tMysqlConnection à la page 2618). Vous n’utiliserez pas cette option pour cet exemple.

5. Dans la liste déroulante Property type, sélectionnez Repository pour que tous les champs de connexion à la base de données soient renseignés automatiquement. Si vous n’avez pas configuré de connexion aux bases de données dans le dossier Metadata du répertoire DB connection, renseignez manuellement ces informations de connexion après avoir sélectionné Built-in.

6. Dans le champ Table Name, saisissez le nom de la nouvelle table à créer.

7. Si vous souhaitez récupérer le schéma d’une métadonnée (sans que celle-ci ne soit une connexion à une base de données), sélectionnez Repository puis la métadonnée souhaitée.

8. Dans tous les cas (Built-in ou Repository), cliquez sur Edit Schema pour vérifier le type de données dans la colonne DB Type. Cliquez sur Edit schema pour définir la structure des données.

9. Cliquez sur le bouton Reset DB Types si la colonne DB type est vide ou affiche des marques de différence (couleur orange). Cela permet de donner à n’importe quel type de données le type de données adéquat. Cliquez sur OK pour valider vos modifications et fermer la boîte de dialogue.


**Résultats**

La table créée est vide mais elle contient toutes les colonnes définies dans le schéma.

```sql
SELECT * FROM 'newcarstable';
```

**Résultat 1**

<table>
<thead>
<tr>
<th>ID_Owners</th>
<th>Reg_Car</th>
<th>Make</th>
<th>Color</th>
<th>ID_Reseller</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


**tCreateTemporaryFile**

Ce composant crée un fichier temporaire dans un répertoire défini. Ce composant permet de conserver ou de supprimer ce fichier temporaire, selon le cas, après l’exécution du Job.

**Propriétés du tCreateTemporaryFile Standard**

Ces propriétés sont utilisées pour configurer le tCreateTemporaryFile s’exécutant dans le framework de Jobs Standard.

Le composant tCreateTemporaryFile Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove file when execution is over</td>
<td>Cochez cette case pour que le fichier temporaire soit supprimé après l’exécution du Job.</td>
</tr>
<tr>
<td>Use default temporary system directory</td>
<td>Cochez cette case afin de créer le fichier dans le répertoire système temporaire par défaut.</td>
</tr>
<tr>
<td>Directory</td>
<td>Spécifiez le répertoire dans lequel créer le fichier temporaire.</td>
</tr>
<tr>
<td></td>
<td>Ce champ est disponible uniquement lorsque la case Use default temporary system directory est décochée.</td>
</tr>
<tr>
<td>Template</td>
<td>Saisissez le nom du fichier temporaire. Ce nom doit contenir les caractères XXXX, par exemple talend_XXXX.</td>
</tr>
<tr>
<td>Suffix</td>
<td>Saisissez une extension de fichier pour le fichier temporaire.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics                       | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILEPATH</td>
<td>chemin où a été créé le fichier. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>
Pour renseigner un champ ou une expression à l'aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>Utilisation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Règle d’utilisation</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liens de sortie (de ce composant à un autre) : <strong>Trigger</strong> : OnSubjobOk, OnSubjobError, Run if, OnComponentOk, OnComponentError.</td>
</tr>
</tbody>
</table>


Pour plus d’informations concernant les liens, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*.

**Scénario : Créer un fichier temporaire et y écrire des données**

Ce scénario décrit un Job créant un fichier temporaire dans un répertoire système temporaire par défaut, écrit des données dans ce fichier puis affiche ces données dans la console.
**Construire le Job**

**Procédure**

1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette : un **tCreateTemporaryFile**, un **tJava**, un **tRowGenerator**, un **tFileOutputDelimited**, un **tFileInputDelimited** et un **tLogRow**.

2. Reliez le **tRowGenerator** au **tFileOutputDelimited** à l’aide d’un lien **Row > Main**.

3. Répétez l’opération pour relier le **tFileInputDelimited** au **tLogRow**.

4. Reliez le **tCreateTemporaryFile** au **tJava** à l’aide d’un lien **Trigger > OnSubjobOk**.

5. Répétez l’opération pour connecter le **tJava** au **tRowGenerator** et le **tRowGenerator** au **tFileInputDelimited**.

**Configurer les composants**

**Créer le fichier temporaire**

**Procédure**

1. Double-cliquez sur le **tCreateTemporaryFile** pour ouvrir sa vue **Basic settings**.
2. Cochez la case **Remove file when execution is over** pour supprimer le fichier temporaire créé après exécution du Job.

3. Cochez la case **Use default temporary system directory** pour créer le fichier dans le répertoire système temporaire par défaut.

4. Dans le champ **Template**, saisissez le nom du fichier temporaire, nom devant contenir les caractères XXXX. Dans cet exemple, saisissez `talend_XXXX`.

5. Dans le champ **Suffix**, saisissez l'extension du fichier temporaire. Dans cet exemple, saisissez `dat`.

6. Double-cliquez sur le **Java** pour ouvrir sa vue **Basic settings**.

7. Dans le champ **Code**, saisissez le code suivant pour afficher dans la console le répertoire système temporaire par défaut et le chemin du fichier temporaire qui sera créé :

   ```java
   System.out.println("The default system temporary directory is:" + (String)System.getProperty("java.io.tmpdir"));
   System.out.println("The path to the temporary file is:" + (String)globalMap.get("tCreateTemporaryFile_1_FILEPATH"));
   ```

**Écrire les données dans le fichier**

**Procédure**

1. Double-cliquez sur le **RowGenerator** pour ouvrir l’éditeur **RowGenerator Editor**.
2. Cliquez sur le bouton [+] pour ajouter deux colonnes : id de type Integer et name de type String. Dans la colonne Functions, sélectionnez la fonction prédéfinie Numeric.sequence(String,int,int) pour id et TalendDataGenerator.getFirstName() pour name.

3. Dans le champ Number of Rows for RowGenerator, saisissez 5 pour générer cinq lignes.

4. Cliquez sur OK pour valider les modifications. Acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

5. Double-cliquez sur le tFileOutputDelimited pour ouvrir sa vue Basic settings.

6. Dans le champ File Name, appuyez sur les touches Ctrl+Espace et, dans la liste des variables globales qui s’affiche, sélectionnez ((String)globalMap.get("tCreateTemporaryFile_1_FILEPATH")).

Lire les données du fichier

Procédure

1. Double-cliquez sur le tFileInputDelimited pour ouvrir sa vue Basic settings.
2. Dans le champ **File name/Stream**, appuyez sur les touches **Ctrl+Espace** et, dans la liste des variables globales affichée, sélectionnez `((String)globalMap.get("tCreateTemporaryFile_1_FILEPATH"))`.

3. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans la fenêtre qui s’ouvre, définissez le schéma en ajoutant deux colonnes : *id* de type **Integer** et *name* de type **String**.

4. Cliquez sur **OK** pour valider les modifications et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

5. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

6. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.
Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder le Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour exécuter le Job.

```
[statistics] connecting to socket on port 3406
[statistics] connected
The default system temporary directory is:
C:\Users\lena_l\AppData\Local\Temp\nThe path to the temporary file is:
C:\Users\lena_l\AppData\Local\Temp\talend_MHTI.dat
+----------
| tLogRow_1 |
+----------
| id | name |
+----------
| 1 | Franklin |
| 2 | Lyndon |
| 3 | Richard |
| 4 | Thomas |
| 5 | Theodore |
+----------
[statistics] disconnected
```

Le fichier talend_MHTI.dat est créé dans le répertoire système temporaire par défaut C:\Users\lena_li\AppData\Local\Temp\ durant l'exécution du Job. Les cinq lignes de données générées sont écrites dans ce fichier, puis le fichier est supprimé après exécution du Job.
tDB2BulkExec

Ce composant exécute une action Insert sur les données fournies et améliore les performances des opérations d'Insert dans une base de données DB2.

Propriétés du tDB2BulkExec Standard

Ces propriétés sont utilisées pour configurer le tDB2BulkExec s'exécutant dans le framework de Jobs Standard.

Le composant tDB2BulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
</tbody>
</table>

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.

Remarque :

Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.
Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

| Host | Adresse IP du serveur de base de données. |
| Port | Numéro du port d’écoute du serveur de base de données. |
| Database | Nom de la base de données. |
| Table Schema | Nom du schéma. |
| Username et Password | Informations d’authentification sur l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Table | Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois. |
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  

- **None** : n’effectuer aucune opération de table.  
- **Drop and create the table** : supprimer la table puis en créer une nouvelle.  
- **Create a table** : créer une table qui n’existe pas encore.  
- **Create table if doesn’t exist** : créer la table si nécessaire.  
- **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.  
- **Clear a table** : supprimer le contenu de la table. |
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  

- **Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.  

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  

- **View schema** : sélectionnez cette option afin de voir le schéma.
<table>
<thead>
<tr>
<th>Use Ingest Command</th>
<th>Use Ingest Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
<td></td>
</tr>
<tr>
<td>Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load From</th>
<th>Load From</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez la source des données à ajouter.</td>
<td></td>
</tr>
<tr>
<td>• FILE : charge des données d'un fichier.</td>
<td></td>
</tr>
<tr>
<td>• PIPE : charge des données d'un tube.</td>
<td></td>
</tr>
<tr>
<td>• FOLDER : chargement des données de différents fichiers dans un dossier.</td>
<td></td>
</tr>
<tr>
<td>Cette liste est disponible uniquement lorsque la case Use Ingest Command est cochée.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data File</th>
<th>Data File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nom du fichier à charger.</td>
<td></td>
</tr>
<tr>
<td>Avertissement :</td>
<td></td>
</tr>
<tr>
<td>Le fichier est situé sur la machine spécifiée par l'URI dans le champ Host et doit être sur la même machine que le serveur de la base de données.</td>
<td></td>
</tr>
<tr>
<td>Ce champ est disponible uniquement lorsque l'option PIPE ou FOLDER est sélectionnée dans la liste Load From.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pipe Name</th>
<th>Pipe Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisissez le nom du tube.</td>
<td></td>
</tr>
<tr>
<td>Ce champ est disponible uniquement lorsque l’option PIPE est sélectionnée dans la liste Load From.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Folder</th>
<th>Folder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spécifiez le chemin d'accès au dossier contenant les fichiers à charger.</td>
<td></td>
</tr>
<tr>
<td>Ce champ est disponible uniquement lorsque l’option FOLDER est sélectionnée dans la liste Load From.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action on Data</th>
<th>Action on Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :</td>
<td></td>
</tr>
<tr>
<td>• Insert : ajouter de nouveaux enregistrements à la table. Le Job s'arrête lorsqu'il détecte des doublons.</td>
<td></td>
</tr>
</tbody>
</table>
- **Replace** : ajouter de nouveaux enregistrements à la table. Si un ancien enregistrement de la table a la même valeur qu’un nouvel enregistrement pour l’index PRIMARY KEY ou UNIQUE, l’ancien enregistrement est supprimé avant l’insertion du nouveau.
- **Update** : mettre à jour les enregistrements existants.
- **Delete** : supprimer les enregistrements correspondant aux données d’entrée.
- **Merge** : fusionner les données d’entrée dans la table.

Les options **Delete** et **Merge** sont disponibles uniquement lorsque la case **Use Ingest Command** est cochée.

<table>
<thead>
<tr>
<th><strong>File Glob Pattern</strong></th>
<th>Spécifiez l’expression globale pour les fichiers à charger. Ce champ est disponible uniquement lorsque l’option FOLDER est sélectionnée dans la liste déroulante Load From.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Where Clause</strong></td>
<td>Saisissez la clause WHERE afin de filtrer les données à traiter. Ce champ est disponible uniquement lorsque l’option Update ou Delete est sélectionnée dans la liste Action on Data.</td>
</tr>
<tr>
<td><strong>Custom Insert Values Clause</strong></td>
<td>Cochez cette case et, dans le champ Insert Values Clause qui s’affiche, saisissez la clause VALUES pour l’opération Insert. Cette case est disponible uniquement lorsque la case Use Ingest Command est cochée et que l’option Insert est sélectionnée dans la liste Action on Data.</td>
</tr>
<tr>
<td><strong>Custom Update Set Clause</strong></td>
<td>Cochez cette case et spécifiez la clause SET pour l’opération Update, en renseignant la table Set Mapping. Cette case est disponible uniquement lorsque la case Use Ingest Command est cochée et que l’option update est sélectionnée dans la liste Action on Data.</td>
</tr>
</tbody>
</table>
| **Set Mapping**       | Renseignez cette table afin de spécifier la clause SET pour l’opération Update.  
  - **Column** : nom de la colonne. Par défaut, les champs de la colonne Column sont les mêmes que dans le schéma.  
  - **Expression** : expression pour la colonne correspondante. Cette table est disponible uniquement lorsque la case Custom Update Set Clause est cochée. |
| **Merge Clause**      | Spécifiez la clause MERGE pour l’opération Merge. |
Cette table est disponible uniquement lorsque la case **Use Ingest Command** est cochée et que l’option **merge** est sélectionnée dans la liste **Action on Data**.

<table>
<thead>
<tr>
<th>Content Format</th>
<th>Sélectionnez le format du fichier d’entrée, <strong>Delimited</strong> ou <strong>Positional</strong>. Cette liste est disponible uniquement lorsque la case <strong>Use Ingest Command</strong> est cochée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delimited By</td>
<td>Saisissez le caractère de séparation des champs dans le fichier délimité. Ce champ est disponible uniquement lorsque l’option <strong>Delimited</strong> est sélectionnée dans la liste <strong>Content Format</strong>.</td>
</tr>
<tr>
<td>Optionally Enclosed By</td>
<td>Saisissez le caractère entourant les chaînes de caractères dans le fichier délimité. Ce champ est disponible uniquement lorsque l’option <strong>Delimited</strong> est sélectionnée dans la liste <strong>Content Format</strong>.</td>
</tr>
<tr>
<td>Fixed Length</td>
<td>Saisissez la longueur (en octets) de l’enregistrement du fichier positionnel. Ce champ est disponible uniquement lorsque l’option <strong>Positional</strong> est sélectionnée dans la liste <strong>Content Format</strong>.</td>
</tr>
</tbody>
</table>

**Mapping**

Renseignez cette table afin de spécifier les relations de mapping entre la colonne source et la colonne de la table DB2.

- **Column** : nom de la colonne. Par défaut, les champs de la colonne **Column** sont les mêmes que dans le schéma.
- **Is Table Column** : cochez cette case si la colonne correspondante est une colonne de table.
- **Start Position** : position de départ de la colonne correspondante.
- **End Position** : position finale de la colonne correspondante.

Les colonnes **Start Position** et **End Position** sont disponibles uniquement lorsque l’option **Positional** est sélectionnée dans la liste **Content Format**. Cette table est disponible uniquement lorsque la case **Use Ingest Command** est cochée.

| Script Generated Folder | Spécifiez le répertoire dans lequel le fichier de script sera créé. Ce champ est disponible uniquement lorsque la case **Use Ingest Command** est cochée. |

**Advanced settings**

| Additional JDBC parameters | Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. |
**Remarque :**
Vous pouvez configurer les paramètres d’encodage dans ce champ.

<table>
<thead>
<tr>
<th><strong>Field terminated by</strong></th>
<th>Caractère, chaîne ou expression régulière séparant les champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Date Format</strong></td>
<td>Utilisez ce champ pour définir l’ordre des mois et des jours.</td>
</tr>
<tr>
<td><strong>Time Format</strong></td>
<td>Utilisez ce champ pour définir l’ordre des heures, minutes et secondes.</td>
</tr>
<tr>
<td><strong>Timestamp Format</strong></td>
<td>Utilisez ce champ pour définir l’ordre de la date et de l’heure.</td>
</tr>
<tr>
<td><strong>Remove Load Pending</strong></td>
<td>Quand la case est cochée, débloque la table en état “Pending”, après un chargement massif.</td>
</tr>
<tr>
<td><strong>Load options</strong></td>
<td>Cliquez sur le bouton [+] afin d’ajouter des options de chargement des données.</td>
</tr>
<tr>
<td></td>
<td><strong>Parameter</strong> : dans la liste, sélectionnez un paramètre de chargement.</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong> : saisissez une valeur pour le paramètre sélectionné.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| **Global Variables** | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable *After* et retourne un entier. |
| | **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable *After* et retourne un entier. |
| | **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable *After* et retourne un entier. |
| | **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable *After* et retourne un entier. |
| | **QUERY** : requête traitée. Cette variable est une variable *After* et retourne une chaîne de caractères. |
| | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. |
| | Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. |
| | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* |
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes DB2.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé au composant tDB2BulkExec, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du tMysqlOutputBulkExec.**
- **Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du tOracleBulkExec.**
tDB2Close

Ce composant ferme une connexion à la base de données DB2 connectée.

**Propriétés du tDB2Close Standard**

Ces propriétés sont utilisées pour configurer le tDB2Close s'exécutant dans le framework de Jobs Standard.

Le composant tDB2Close Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tDB2Connection dans la liste.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tDB2Commit

Ce composant commite en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

Le tDB2Commit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tDB2Commit Standard

Ces propriétés sont utilisées pour configurer le tDB2Commit s’exécutant dans le framework de Jobs Standard.

Le composant tDB2Commit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tDB2Connection dans la liste.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
<tr>
<td></td>
<td><strong>Avertissement :</strong> Si vous utilisez un lien de type Row &gt; Main pour relier le tDB2Commit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants DB2, notamment le \texttt{tDB2Connection} et le \texttt{tDB2Rollback}.</th>
</tr>
</thead>
</table>
| Dynamic settings   | Cliquez sur le bouton \([+]\) pour ajouter une ligne à la table. Dans le champ \texttt{Code}, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table \texttt{Dynamic settings} est disponible uniquement lorsque la case \texttt{Use an existing connection} est cochée dans la vue \texttt{Basic settings}. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le \textit{Guide utilisateur du Studio Talend}. |

### Scénario associé

Pour un scénario associé au composant \texttt{tDB2Commit}, consultez \textit{Scénario : Insérer des données dans des tables mère/fille} à la page 2620.
tDB2Connection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Propriétés du tDB2Connection Standard

Ces propriétés sont utilisées pour configurer le tDB2Connection s’exécutant dans le framework de Jobs Standard.

Le composant tDB2Connection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

Remarque : Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Table Schema</strong></td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez <strong>Custom</strong> et définissez-le manuellement. Ce</td>
</tr>
</tbody>
</table>
### champ est obligatoire pour la manipulation des données des bases de données.

| **Use or register a shared DB Connection** | **Use or register a shared DB Connection**<br>Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.<br>Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.<br>Cette case est indisponible lorsque la case **Specify a data source alias** est cochée. |
| **Specify a data source alias** | **Specify a data source alias**<br>Cochez cette case et spécifiez l’alias de la source de données créée dans Talend Runtime pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647.<br>Cette case n’est pas disponible lorsque la case **Use or register a shared DB Connection** est cochée. |
| **Data source alias** | **Data source alias**<br>Saisissez l’alias de la source de données créée du côté de Talend Runtime.<br>Ce champ est disponible uniquement lorsque la case **Specify a data source alias** est cochée. |

### Advanced settings

| **Additional JDBC parameters** | **Additional JDBC parameters**<br>Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez.<br>**Remarque :**<br>Vous pouvez configurer les paramètres d’encodage dans ce champ. |
| **Auto Commit** | **Auto Commit**<br>Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.<br>Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d'auto-commit est désactivée et les modifications doivent être commitées de maniè
Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants DB2, notamment les composants tDB2Commit et tDB2Rollback.</th>
</tr>
</thead>
</table>

### Scénarios associés

Pour un scénario associé au composant tDB2Connection, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
**tDB2Input**

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

Si des guillemets doubles existent dans le nom des colonnes d’une table, les guillemets doubles ne peuvent être récupérés avec les colonnes. Il est donc recommandé de ne pas utiliser de guillemet double dans le nom des colonnes dans les tables de la base de données DB2.

**Propriétés du tDB2Input Standard**

Ces propriétés sont utilisées pour configurer le tDB2Input s’exécutant dans le framework de Jobs Standard.

Le composant tDB2Input Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque** :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque : Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné. Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.
Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur d’annuaire DB2.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
</tbody>
</table>
**Table Name**
Nom de la table source dans laquelle capturer les changements apportés aux données.

**Query type et Query**
Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

**Specify a data source alias**
Cochez cette case et spécifiez l’alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans *Talend Runtime* pour récupérer les données d’une base de données MySQL à la page 2647. Cette option est indisponible lorsque la case *Use an existing connection* est cochée.

**Data source alias**
Saisissez l’alias de la source de données créée du côté de *Talend Runtime*.
Ce champ est disponible uniquement lorsque la case *Specify a data source alias* est cochée.

**Advanced settings**

**Additional JDBC parameters**
Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez.

**Remarque :**
Vous pouvez configurer les paramètres d’encodage dans ce champ.

**Trim all the String/Char columns**
Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.

**Trim column**
Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables

**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>QUERY</td>
<td>Requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données DB2.</td>
</tr>
</tbody>
</table>

Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

**Scénarios associés**

Pour des scénarios associés, consultez :

Consultez également Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520.
tDB2Output

Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.
Le tDB2Output écrit, met à jour, modifie ou supprime les données d’une base de données.

Propriétés du tDB2Output Standard

Ces propriétés sont utilisées pour configurer le tDB2Output s’exécutant dans le framework de Jobs Standard.
Le composant tDB2Output Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

| Property type | Peut être Built-in ou Repository |
**Built-in** : Propriétés utilisées ponctuellement.

**Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.

Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.

Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d'authentification de l'utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td><strong>None</strong> : n'effectuer aucune opération de table.</td>
</tr>
<tr>
<td></td>
<td><strong>Drop and create the table</strong> : supprimer la table puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td><strong>Create a table</strong> : créer une table qui n'existe pas encore.</td>
</tr>
<tr>
<td></td>
<td><strong>Create table if doesn't exist</strong> : créer la table si nécessaire.</td>
</tr>
<tr>
<td></td>
<td><strong>Drop a table if exists and create</strong> : supprimer la table si elle existe déjà, puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td><strong>Clear a table</strong> : supprimer le contenu de la table.</td>
</tr>
<tr>
<td></td>
<td><strong>Truncate table</strong> : supprimer le contenu de la table. Il n'y a pas de possibilité de rollback.</td>
</tr>
<tr>
<td></td>
<td><strong>Truncate table with reuse storage</strong> : supprimer le contenu de la table. Il n'y a pas de possibilité de rollback. Cependant, vous pouvez réutiliser le stockage alloué à la table, même si le stockage est considéré comme vide.</td>
</tr>
</tbody>
</table>

**Avertissement** :
Si vous cochez la case **Use an existing connection** et que vous sélectionnez **Truncate table** ou **Truncate table with reuse storage** dans la liste **Action on table**, une instruction...
de commit est invoquée avant l’opération TRUNCATE, car l’instruction TRUNCATE doit être la première instruction dans une transaction.

| Action on data | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :
|                | **Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
| Action on data | **Update** : Mettre à jour les entrées existantes.
|                | **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
| Action on data | **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
| Action on data | **Delete** : Supprimer les entrées correspondantes au flux d’entrée.

⚠️ **Avertissement** :
Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update et Delete**. Pour cela, cliquez sur le bouton […] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
| Schema et Edit schema | **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.
Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

---

### Specify a data source alias

Cochez cette case et spécifiez l’alias de la source de données créée dans **Talend Runtime** pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans **Talend Runtime**. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647. Cette option est indisponible lorsque la case **Use an existing connection** est cochée.

### Data source alias

Saisissez l’alias de la source de données créée du côté de **Talend Runtime**.

Ce champ est disponible uniquement lorsque la case **Specify a data source alias** est cochée.

### Die on error

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

---

### Advanced settings

### Additional JDBC parameters

Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez.

**Remarque** :

Vous pouvez configurer les paramètres d’encodage dans ce champ.
| Commit every | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de transaction (cependant pas de rollback) et surtout une meilleure performance des exécutions. |
| Additional Columns | Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier. |
| Name | Saisissez le nom de la colonne à modifier ou à insérer. |
| SQL expression | Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes. |
| Position | Sélectionnez Before, Replace ou After, en fonction de l’action à effectuer sur la colonne de référence. |
| Reference column | Saisissez une colonne de référence que le composant tDB2Output peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier. |
| Use field options | Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données. |
| Convert columns and table names to uppercase | Cochez cette case pour mettre le nom des colonnes et celui de la table en majuscules. |
| Enable debug mode | Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données. |
| Support null in “SQL WHERE” statement | Cochez cette case pour prendre en compte les valeurs Null d’une table de base de données. |
| **Remarque** | Assurez-vous que la case Nullable est bien cochée pour les colonnes du schéma correspondantes. |
| Use Batch | Cochez cette case pour activer le mode de traitement par lots pour le traitement des données. |
| Batch Size | Spécifiez le nombre d’enregistrements à traiter dans chaque lot. Ce champ est disponible uniquement lorsque la case Use batch mode est cochée. |
| tStat Catcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong></td>
<td>Nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong></td>
<td>Nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong></td>
<td>Nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_REJECTED</strong></td>
<td>Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>QUERY</strong></td>
<td>Requête traitée. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

**Règle d'utilisation**

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données DB2. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exem...
ple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Scénarios associés**

Pour un scénario associé au composant **tDB2Output**, consultez :

- **Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput** à la page 2667.
tDB2Rollback

Ce composant évite le commit de transaction involontaire.
Le tDB2Rollback annule la transaction dans une base de données connectée.

Propriétés du tDB2Rollback Standard

Ces propriétés sont utilisées pour configurer le tDB2Rollback s’exécutant dans le framework de Jobs Standard.
Le composant tDB2Rollback Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

| Database | Sélectionnez un type de base de données dans la liste et cliquez sur Apply. |
| Component list | Sélectionnez le composant de connexion tDB2Connection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours. |
| Close Connection | Décrochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé en association avec des composants DB2, notamment avec le tDB2Connection et le tDB2Commit. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant |
la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Pour un scénario associé au tDB2Rollback, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623 du composant tMySQLRollback.
tDB2Row

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler), selon la nature de la requête et de la base de données. Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Le tDB2Row est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

**Propriétés du tDB2Row Standard**

Ces propriétés sont utilisées pour configurer le tDB2Row s'exécutant dans le framework de Jobs Standard.

Le composant tDB2Row Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie. <strong>Remarque :</strong> Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez : 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue <strong>Basic settings</strong> du composant de connexion créant cette connexion. 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métdonnée du schéma dans la fenêtre Repository Content.</td>
</tr>
<tr>
<td>Table Name</td>
<td>Nom de la table à traiter.</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>Query type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez la requête appropriée dans le Repository. Le champ <strong>Query</strong> est renseigné automatiquement.</td>
</tr>
<tr>
<td>Query</td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td>Specify a data source alias</td>
<td>Cochez cette case et spécifiez l’alias de la source de données créée dans <em>Talend Runtime</em> pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans <em>Talend Runtime</em>. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans <em>Talend Runtime</em> pour récupérer les données d’une base de données MySQL à la page 2647. Cette option est indisponible lorsque la case <strong>Use an existing connection</strong> est cochée.</td>
</tr>
<tr>
<td>Data source alias</td>
<td>Saisissez l’alias de la source de données créée du côté de <em>Talend Runtime</em>. Ce champ est disponible uniquement lorsque la case <strong>Specify a data source alias</strong> est cochée.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien <strong>Row &gt; Rejects</strong>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| Additional JDBC parameters | Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. |
|                          | **Remarque** : |
|                          | Vous pouvez configurer les paramètres d’encodage dans ce champ. |
| Propagate QUERY’s recordset | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. |
|                          | **Remarque** : |
|                          | Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le... |
Le résultat de la requête doit être de type `Object`. Ce composant est généralement suivi du `tParseRecordSet`.

### Commit every
Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.

### Use PreparedStatement

- **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL.
- **Parameter Type** : Saisissez le type du paramètre.
- **Parameter Value** : Saisissez la valeur du paramètre.

**Remarque** :
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**QUERY** : requête traitée. Cette variable est une variable `Flow` et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case `Die on error` est décochée, si le composant a cette option.

Une variable `Flow` fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le `Guide utilisateur du Studio Talend`.

### Utilisation

**Règle d’utilisation**
Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.
Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


### Scénarios associés

Pour un scénario associé au composant tDB2Row, consultez :

- Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
- Procédure du composant tDBSQLRow.
- Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du composant tMysqlRow.
tDB2SCD

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée. Le tDB2SCD reflète et traque les modifications d’une table DB2 SCD dédiée.

Propriétés du tDB2SCD Standard

Ces propriétés sont utilisées pour configurer le tDB2SCD s’exécutant dans le framework de Jobs Standard.

Le composant tDB2SCD Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
</table>

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà défini. |

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

| Property type | Le schéma peut être Built-in ou distant dans le Repository. |
**Built-in** : Propriétés utilisées ponctuellement.

**Repository** : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.

| **Host** | Adresse IP du serveur de base de données. |
| **Port** | Numéro du port d'écoute du serveur de base de données. |
| **Database** | Nom de la base de données. |
| **Table schema** | Nom du schéma de la base de données. |

**Username et Password**
Informations d'authentification de l'utilisateur de la base de données.
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Table**
Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.

**Schema et Edit schema**
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

**Repository** : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.
**SCD Editor**

L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d’informations, consultez Méthodologie de gestion du SCD à la page 2716.

**Use memory saving Mode**

Cochez cette case pour améliorer les performances du système.

**Die on error**

Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Additional JDBC parameters</strong></th>
<th>Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Vous pouvez configurer les paramètres d’encodage dans ce champ.</td>
</tr>
</tbody>
</table>

**End date time details**

Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est *12:00:00*.

Ce champ apparaît uniquement lorsqu’un SCD de Type 2 est utilisé et lorsque *Fixed year value* est sélectionné pour créer la date de fin du SCD.

**Debug mode**

Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Variables globales

<table>
<thead>
<tr>
<th><strong>Global Variables</strong></th>
<th><strong>NB_LINE_UPDATED</strong> : nombre de lignes mises à jour. Cette variable est une variable <em>After</em> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>NB_LINE_INSERTED</strong> : nombre de lignes insérées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_LINE_REJECTED</strong> : nombre de lignes rejetées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <em>Flow</em> fonctionne durant l’exécution d’un composant. Une variable <em>After</em> fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Limitation</td>
<td>Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé, consultez tMysqlSCD à la page 2712.
tDB2SCDELT

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table DB2 SCD dédiée.

Propriétés du tDB2SCDELT Standard

Ces propriétés sont utilisées pour configurer le tDB2SCDELT s’exécutant dans le framework de Jobs Standard.

Le composant tDB2SCDELT Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.
Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend.*

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <em>Password,</em> puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Source table</td>
<td>Nom de la table contenant les données à filtrer.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois pour que l’opération d’insert soit autorisée.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
  - **None** : n’effectuer aucune opération de table.  
  - **Drop and create the table** : supprimer la table puis en créer une nouvelle.  
  - **Create a table** : créer une table qui n’existe pas encore.  
  - **Create table if doesn’t exist** : créer la table si nécessaire.  
  - **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.  
  - **Clear a table** : supprimer le contenu de la table.  
  - **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
  Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository,* trois options sont disponibles :  
  - **View schema** : sélectionnez cette option afin de voir le schéma.  
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.  
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans |
le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


### Surrogate Key
Sélectionnez dans la liste une colonne à utiliser comme clé de substitution.

### Creation
Sélectionnez la méthode à utiliser pour générer la clé de substitution.

Pour plus d’informations concernant les méthodes de création, consultez Méthodologie de gestion du SCD à la page 2716.

### Source Keys
Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l’unicité des données entrantes.

### Source fields value include Null
Cochez cette case pour autoriser les valeurs Nulls dans les colonnes source.

**Remarque :**

Les colonnes source font référence aux champs définis dans les tables SCD type 1 fields et SCD type 2 fields.

### Use SCD Type 1 fields
Utilisez le type 1 si vous n’avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.

### Use SCD Type 2 fields
Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.

**Start date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l’une des colonnes d’entrée du schéma comme date de départ (**Start Date**) dans la table SCD.

**End Date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne **End Date** a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l’option **Fixed Year value** et saisir une année fictive.

**Log Active Status** : Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut **true** et **false**.
Cette colonne permet de repérer facilement le journal actif.

**Log versions** : Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.

### Advanced settings

<table>
<thead>
<tr>
<th>Additional JDBC parameters</th>
<th>Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque</strong> :</td>
</tr>
<tr>
<td></td>
<td>Vous pouvez configurer les paramètres d’encodage dans ce champ.</td>
</tr>
<tr>
<td>Debug mode</td>
<td>Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type <strong>Row Main</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exem</td>
</tr>
</tbody>
</table>
Scénario associé

Pour un scénario associé, consultez tDB2SCD à la page 621 et tMysqlSCD à la page 2712.
tDB2SP

Ce composant appelle des procédure stockée de base de données.

Propriétés du tDB2SP Standard

Ces propriétés sont utilisées pour configurer le tDB2SP s'exécutant dans le framework de Jobs Standard.

Le composant tDB2SP Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d'authentification sur l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il est réutilisable. Voir également le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
<tr>
<td><strong>SP Name</strong></td>
<td>Saisissez le nom exact de la Procédure Stockée (SP).</td>
</tr>
<tr>
<td><strong>Is Function / Return result in</strong></td>
<td>Cochez cette case, si une seule valeur doit être retournée. Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.</td>
</tr>
</tbody>
</table>
### Parameters

Cliquez sur le bouton [+] et sélectionnez dans le champ **Schema Columns** les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure.

Sélectionnez le **Type** de paramètre :

- **IN** : paramètre d'entrée (Input)
- **OUT** : paramètre de sortie (Output)/valeur retournée
- **IN OUT** : les paramètres d'entrée doivent être retournées sous forme de valeur, même après modifications via la procédure (fonction).
- **RECORDSET** : les paramètres d'entrée doivent être retournées sous forme d'ensemble de valeurs, au lieu d'une valeur unique.

**Remarque :**
Consultez **Scénario : Insérer des données dans des tables mère/fille** à la page 2620 si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.

### Specify a data source alias

Cochez cette case et spécifiez l'alias de la source de données créée dans **Talend Runtime** pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans **Talend Runtime**. Pour voir un exemple de cas d'utilisation, consultez **Scénario : Déploiement de votre Job dans Talend Runtime** pour récupérer les données d'une base de données MySQL à la page 2647.

Cette option est indisponible lorsque la case **Use an existing connection** est cochée.

### Data source alias

Saisissez l'alias de la source de données créée du côté de **Talend Runtime**.

Ce champ est disponible uniquement lorsque la case **Specify a data source alias** est cochée.

### Advanced settings

#### Additional JDBC parameters

Spécifiez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez.

**Remarque :**
Vous pouvez configurer les paramètres d’encodage dans ce champ.

#### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.
## Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d’entrée sont autorisés.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  
La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lors qu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.  

### Scénario associé

Pour un scénario associé, consultez

- Scénario : Récupérer des informations personnelles à l’aide d’une procédure stockée à la page 2594.
- Scénario : Utiliser le tMysqlSP pour trouver le libellé State à l’aide d’une procédure stockée à la page 2734.
- Scénario : Vérifier le format de numéros à l’aide d’une procédure stockée à la page 2976.
- Scénario : Exécuter une procédure stockée à l’aide du tMDMSP à la page 2320.

Consultez également Scénario : Insérer des données dans des tables mère/fille à la page 2620 si vous voulez analyser un ensemble d’enregistrements d’une table de données ou d’une requête SQL.
Composants de bases de données dynamiques

Talend fournit de nombreux composants de bases de données vous permettant de changer dynamiquement le type de base de données à utiliser. Ces composants sont disponibles dans le groupe **Database Common**, sous la famille **Databases** de la Palette, pour les Jobs standard d’intégration de données.

Chacun de ces composants contient une propriété unique, la liste **Database**, dans sa vue **Basic settings**, dans laquelle vous pouvez sélectionner le type de base de données à utiliser.

Pour plus d’informations concernant ces composants de bases de données dynamiques, consultez :

- tDBBulkExec à la page 635
- tDBClose à la page 637
- tDBColumnList à la page 639
- tDBCommit à la page 640
- tDBConnection à la page 642
- tDBInput à la page 644
- tDBLastInsertId à la page 646
- tDBOutput à la page 647
- tDBOutputBulk à la page 649
- tDBOutputBulkExec à la page 650
- tDBRollback à la page 651
- tDBRow à la page 653
- tDBSCD à la page 655
- tDBSCDELT à la page 656
- tDBSP à la page 657
- tDBTableList à la page 658
tDBBulkExec

Ce composant offre un gain de performance lors de l’exécution d’opérations d’insertion dans une base de données.

Ce composant fonctionne avec différentes bases de données.

Les composants tDBOutputBulk et tDBBulkExec sont généralement utilisés ensemble en tant que parties d’un processus de deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la seconde étape, ce fichier est utilisé dans une instruction INSERT pour alimenter une base de données du type sélectionné. Ces deux étapes sont fusionnées dans le composant tDBOutputBulkExec, détaillé dans une section séparée. L’avantage d’utiliser deux composant séparés réside dans le fait que les données peuvent être transformées avant d’être chargées dans la base de données.

Propriétés du tDBBulkExec Standard

Ces propriétés sont utilisées pour configurer le tDBBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tDBBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Access (tAccessBulkExec à la page 64)
- Amazon (tRedshiftBulkExec à la page 3296)
- Greenplum (tGreenplumBulkExec à la page 1388)
- IBM DB2 (tDB2BulkExec à la page 588)
- Informix (tInformixBulkExec à la page 1811)
- Ingres (tIngresBulkExec à la page 1858)
- Microsoft SQL Server (tMSSqlBulkExec à la page 2533)
- MySQL (tMysqlBulkExec à la page 2602)
- Netezza (tNetezzaBulkExec à la page 2829)
- Oracle (tOracleBulkExec à la page 2908)
- ParAccel (tParAccelBulkExec à la page 3049)
- PostgreSQL (tPostgresqlBulkExec à la page 2829)
- PostgresPlus (tPostgresPlusBulkExec à la page 3198)
- Sybase (ASE et IQ) (tSybaseBulkExec à la page 4024)
• Sybase IQ (tSybaseIQBulkExec à la page 4041)
• Vertica (tVerticaBulkExec à la page 4205)
tDBClose

Ce composant ferme une connexion à une base de données connectée.
Ce composant fonctionne avec différentes bases de données.

Propriétés du tDBClose Standard

Ces propriétés sont utilisées pour configurer le tDBClose s’exécutant dans le framework de Jobs Standard.
Le composant tDBClose Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Access (tAccessClose à la page 68)
- Amazon Aurora (tAmazonAuroraClose à la page 136)
- Amazon Mysql (tAmazonMysqlClose à la page 180)
- Amazon Oracle (tAmazonOracleClose à la page 206)
- Amazon Redshift (tRedshiftClose à la page 3312)
- AS400 (tAS400Close à la page 240)
- FireBird (tFirebirdClose à la page 1243)
- Greenplum (tGreenplumClose à la page 1393)
- IBM DB2 (tDB2Close à la page 595)
- Exasol (tEXAClose à la page 945)
- Informix (tInformixClose à la page 1817)
- Ingres (tIngresClose à la page 1863)
- Interbase (tInterbaseClose à la page 1900)
- JDBC (tJDBCClose à la page 1969)
- MemSQL (tMemSQLClose à la page 2352)
- Microsoft SQL Server (tMSSqlClose à la page 2538)
- MySQL (tMysqlClose à la page 2607)
- Netezza (tNetezzaClose à la page 2834)
- Oracle (tOracleClose à la page 2918)
- ParAccel (tParAccelClose à la page 3054)
- PostgreSQL (tPostgresqlClose à la page 3247)
- PostgresPlus (tPostgresPlusClose à la page 3202)
- SAPHana (tSAPHanaClose à la page 3662)
- SQLite (tSQLiteClose à la page 3852)
- Snowflake (tSnowflakeClose à la page 3755)
- Sybase (ASE et IQ) (tSybaseClose à la page 4030)
- Teradata (tTeradataClose à la page 4097)
- Vertica (tVerticaClose à la page 4212)
tDBColumnList

Ce composant effectue une itération sur toutes les colonnes d’une table d’une base de données et liste le nom des colonnes.

Ce composant fonctionne avec différentes bases de données.

Propriétés du tDBColumnList Standard

Ces propriétés sont utilisées pour configurer le tDBColumnList s’exécutant dans le framework de Jobs Standard.

Le composant tDBColumnList Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Microsoft SQL Server (tMSSqlColumnList à la page 2540)
- MySQL (tMysqlColumnList à la page 2609)
tDBCommit

Ce composant valide les données traitées à travers le Job dans la base de données connectée.
Ce composant fonctionne avec différentes bases de données.

Propriétés du tDBCommit Standard

Ces propriétés sont utilisées pour configurer le composant tDBCommit s'exécutant dans le framework de Jobs Standard.
Le composant tDBCommit Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Access (tAccessCommit à la page 70)
- Amazon Aurora (tAmazonAuroraCommit à la page 138)
- Amazon Mysql (tAmazonMysqlCommit à la page 182)
- Amazon Oracle (tAmazonOracleCommit à la page 208)
- AS400 (tAS400Commit à la page 242)
- Amazon Redshift (tRedshiftCommit à la page 3314)
- FireBird (tFirebirdCommit à la page 1245)
- Greenplum (tGreenplumCommit à la page 1395)
- IBM DB2 (tDB2Commit à la page 597)
- Exasol (tEXACcommit à la page 947)
- Informix (tInformixCommit à la page 1819)
- Ingres (tIngresCommit à la page 1865)
- Interbase (tInterbaseCommit à la page 1902)
- JDBC (tJDBCCommit à la page 1973)
- Microsoft SQL Server (tMSSqlCommit à la page 2543)
- MySQL (tMysqlCommit à la page 2615)
- Netezza (tNetezzaCommit à la page 2836)
- Oracle (tOracleCommit à la page 2920)
- ParAccel (tParAccelCommit à la page 3056)
• PostgreSQL (tPostgresqlCommit à la page 3249)
• PostgresPlus (tPostgresPlusCommit à la page 3204)
• SAPHana (tSAPHanaCommit à la page 3663)
• SQLite (tSQLiteCommit à la page 3854)
• Sybase (ASE et IQ) (tSybaseCommit à la page 4032)
• Teradata (tTeradataCommit à la page 4099)
• VectorWise (tVectorWiseCommit à la page 4183)
• Vertica (tVerticaCommit à la page 4214)
tDBConnection

Ce composant ouvre une connexion à une base de données pouvant être réutilisée dans le(s) sous-job(s) suivant(s).

Ce composant fonctionne avec différents bases de données.

Propriétés du tDBConnection Standard

Ces propriétés sont utilisées pour configurer le composant tDBConnection s’exécutant dans le framework de Jobs Standard.

Le composant tDBConnection Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Access (tAccessConnection à la page 72)
- Amazon Aurora (tAmazonAuroraConnection à la page 141)
- Amazon Mysql (tAmazonMysqlConnection à la page 185)
- Amazon Oracle (tAmazonOracleConnection à la page 211)
- Amazon Redshift (tRedshiftConnection à la page 3316)
- AS400 (tAS400Connection à la page 244)
- Exasol (tEXAConnection à la page 950)
- FireBird (tFirebirdConnection à la page 1247)
- Greenplum (tGreenplumConnection à la page 1397)
- IBM DB2 (tDB2Connection à la page 599)
- Informix (tInformixConnection à la page 1821)
- Ingres (tIngresConnection à la page 1867)
- Interbase (tInterbaseConnection à la page 1904)
- JDBC (tJDBCConnection à la page 1975)
- MemSQL (tMemSQLConnection à la page 2354)
- Microsoft SQL Server (tMSSqlConnection à la page 2545)
- MySQL (tMysqlConnection à la page 2618)
- Netezza (tNetezzaConnection à la page 2838)
- Oracle (tOracleConnection à la page 2922)
• ParAccel (tParAccelConnection à la page 3058)
• PostgreSQL (tPostgresqlConnection à la page 3251)
• PostgresPlus (tPostgresPlusConnection à la page 3206)
• SAPHana (tSAPHanaConnection à la page 3665)
• SQLite (tSQLiteConnection à la page 3856)
• Snowflake (tSnowflakeConnection à la page 3757)
• Sybase (ASE et IQ) (tSybaseConnection à la page 4034)
• Teradata (tTeradataConnection à la page 4101)
• VectorWise (tVectorWiseConnection à la page 4185)
• Vertica (tVerticaConnection à la page 4216)
tDBInput

Ce composant extrait des données d'une base de données.
Ce composant fonctionne avec différentes bases de données.

Propriétés du tDBInput Standard

Ces propriétés sont utilisées pour configurer le tDBInput s'exécutant dans le framework de Jobs Standard.
Le composant tDBInput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d'entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d'informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Access (tAccessInput à la page 78)
- Amazon Aurora (tAmazonAuroraInput à la page 144)
- Amazon Mysql (tAmazonMysqlInput à la page 188)
- Amazon Oracle (tAmazonOracleInput à la page 214)
- Amazon Redshift (tRedshiftInput à la page 3319)
- AS400 (tAS400Input à la page 246)
- Exasol (tEXAInput à la page 953)
- FireBird (tFirebirdInput à la page 1250)
- Greenplum (tGreenplumInput à la page 1407)
- IBM DB2 (tDB2Input à la page 602)
- Informix (tInformixInput à la page 1824)
- Ingres (tIngresInput à la page 1870)
- Interbase (tInterbaselInput à la page 1906)
- JDBC (tJDBCInput à la page 1978)
- MemSQL (tMemSQLInput à la page 2357)
- Microsoft SQL Server (tMSSqlInput à la page 2553)
- MySQL (tMysqlInput à la page 2631)
- Netezza (tNetezzaInput à la page 2841)
- Oracle (tOracleInput à la page 2926)
• ParAccel (tParAccelInput à la page 3061)
• PostgreSQL (tPostgresqlInput à la page 3254)
• PostgresPlus (tPostgresPlusInput à la page 3209)
• SAPHana (tSAPHanaInput à la page 3667)
• SAS (tSasInput à la page 3683)
• SQLite (tSQLiteInput à la page 3858)
• Snowflake (tSnowflakeInput à la page 3760)
• Sybase (ASE et IQ) (tSybaseInput à la page 4037)
• Teradata (tTeradataInput à la page 4113)
• VectorWise (tVectorWiseInput à la page 4188)
• Vertica (tVerticalInput à la page 4219)
tDBLastInsertId

Ce composant obtient la valeur de la clé primaire de l’enregistrement ayant été inséré le plus récemment dans une table d’une base de données par un utilisateur.

Ce composant fonctionne avec différentes bases de données.

Propriétés du tDBLastInsertId Standard

Ces propriétés sont utilisées pour configurer le composant tDBLastInsertId s’exécutant dans le framework de Jobs Standard.

Le composant tDBLastInsertId Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- AS400 (tAS400LastInsertId à la page 254)
- Microsoft SQL Server (tMSSqlLastInsertId à la page 2558)
- MySQL (tMysqlLastInsertId à la page 2653)
tDBOutput

Ce composant écrit, met à jour, effectue des modifications ou supprime des entrées dans une base de données.

Ce composant fonctionne avec différentes bases de données.

Propriétés du tDBOutput Standard

Ces propriétés sont utilisées pour configurer le tDBOutput s’exécutant dans le framework de Jobs Standard.

Le composant tDBOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Access (tAccessOutput à la page 82)
- Amazon Aurora (tAmazonAuroraOutput à la page 154)
- Amazon Mysql (tAmazonMysqlOutput à la page 192)
- Amazon Oracle (tAmazonOracleOutput à la page 218)
- Amazon Redshift (tRedshiftOutput à la page 3329)
- AS400 (tAS400Output à la page 256)
- Exasol (tEXAOutput à la page 958)
- FireBird (tFirebirdOutput à la page 1254)
- Greenplum (tGreenplumOutput à la page 1411)
- IBM DB2 (tDB2Output à la page 607)
- Informix (tInformixOutput à la page 1828)
- Ingres (tIngresOutput à la page 1874)
- Interbase (tInterbaseOutput à la page 1910)
- JDBC (tJDBCOutput à la page 1983)
- MemSQL (tMemSQLOutput à la page 2368)
- Microsoft SQL Server (tMSSqlOutput à la page 2561)
- MySQL (tMysqlOutput à la page 2661)
- Netezza (tNetezzaOutput à la page 2853)
- Oracle (tOracleOutput à la page 2935)
• ParAccel (tParAccelOutput à la page 3065)
• PostgreSQL (tPostgresqlOutput à la page 3259)
• PostgresPlus (tPostgresPlusOutput à la page 3213)
• SAPHana (tSAPHanaOutput à la page 3671)
• SAS (tSasOutput à la page 3686)
• SQLite (tSQLiteOutput à la page 3864)
• Snowflake (tSnowflakeOutput à la page 3769)
• Sybase (ASE et IQ) (tSybaseOutput à la page 4059)
• Teradata (tTeradataOutput à la page 4122)
• VectorWise (tVectorWiseOutput à la page 4192)
• Vertica (tVerticaOutput à la page 4223)
tDBOutputBulk

Ce composant écrit un fichier avec des colonnes, en se basant sur le séparateur défini et les standards du type de base de données sélectionné.

Ce composant fonctionne avec différentes bases de données.

Les composants tDBOutputBulk et tDBBulkExec sont généralement utilisés ensemble en tant que parties d’un processus de deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la seconde étape, ce fichier est utilisé dans une instruction INSERT pour alimenter une base de données du type sélectionné. Ces deux étapes sont fusionnées dans le composant tDBOutputBulkExec, détaillé dans une section séparée. L’avantage d’utiliser deux composant séparés réside dans le fait que les données peuvent être transformées avant d’être chargées dans la base de données.

Propriétés du tDBOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tDBOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tDBOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Access (tAccessOutputBulk à la page 88)
- Amazon Redshift (tRedshiftOutputBulk à la page 3336)
- Greenplum (tGreenplumOutputBulk à la page 1417)
- Informix (tInformixOutputBulk à la page 1834)
- Ingres (tIngresOutputBulk à la page 1880)
- Microsoft SQL Server (tMSSqlOutputBulk à la page 2569)
- MySQL (tMysqlOutputBulk à la page 2682)
- Oracle (tOracleOutputBulk à la page 2943)
- ParAccel (tParAccelOutputBulk à la page 3071)
- PostgreSQL (tPostgresqlOutputBulk à la page 3266)
- PostgresPlus (tPostgresPlusOutputBulk à la page 3219)
- Sybase (ASE et IQ) (tSybaseOutputBulk à la page 4065)
- Vertica (tVerticaOutputBulk à la page 4231)
tDBOutputBulkExec

Ce composant exécute l’action d’insertion dans une base de données.

Ce composant fonctionne avec différentes bases de données.

Les composants tDBOutputBulk et tDBBulkExec sont généralement utilisés ensemble en tant que parties d’un processus de deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la seconde étape, ce fichier est utilisé dans une instruction INSERT pour alimenter une base de données du type sélectionné. Ces deux étapes sont fusionnées dans le composant tDBOutputBulkExec, détaillé dans une section séparée. L’avantage d’utiliser deux composant séparés réside dans le fait que les données peuvent être transformées avant d’être chargées dans la base de données.

Propriétés du tDBOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le composant tDBOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tDBOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Access (tAccessOutputBulk à la page 88)
- Amazon Redshift (tRedshiftOutputBulk à la page 3336)
- Greenplum (tGreenplumOutputBulk à la page 1417)
- Informix (tInformixOutputBulk à la page 1834)
- Ingres (tIngresOutputBulk à la page 1880)
- Microsoft SQL Server (tMSSqlOutputBulk à la page 2569)
- MySQL (tMysqlOutputBulk à la page 2682)
- Oracle (tOracleOutputBulk à la page 2943)
- ParAccel (tParAccelOutputBulk à la page 3071)
- PostgreSQL (tPostgresqlOutputBulk à la page 3266)
- PostgresPlus (tPostgresPlusOutputBulk à la page 3219)
- Sybase (ASE et IQ) (tSybaseOutputBulk à la page 4065)
- Vertica (tVerticaOutputBulk à la page 4231)
tDBRollback

Ce composant annule un commit dans une base de données connectée, afin d'éviter les commits involontaires de transactions.

Ce composant fonctionne avec différentes bases de données.

Propriétés du tDBRollback Standard

Ces propriétés sont utilisées pour configurer le composant tDBRollback s'exécutant dans le framework de Jobs Standard.

Le composant tDBRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d'entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d'informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Access (tAccessRollback à la page 95)
- Amazon Aurora (tAmazonAuroraRollback à la page 162)
- Amazon Mysql (tAmazonMysqlRollback à la page 199)
- Amazon Oracle (tAmazonOracleRollback à la page 225)
- Amazon Redshift (tRedshiftRollback à la page 3346)
- AS400 (tAS400Rollback à la page 262)
- Exasol (tEXARollback à la page 965)
- FireBird (tFirebirdRollback à la page 1260)
- Greenplum (tGreenplumRollback à la page 1424)
- IBM DB2 (tDB2Rollback à la page 614)
- Informix (tInformixRollback à la page 1843)
- Ingres (tIngresRollback à la page 1890)
- Interbase (tInterbaseRollback à la page 1916)
- JDBC (tJDBCRollback à la page 1990)
- Microsoft SQL Server (tMSSqlRollback à la page 2577)
- MySQL (tMysqlRollback à la page 2694)
- Netezza (tNetezzaRollback à la page 2859)
- Oracle (tOracleRollback à la page 2953)
- ParAccel (tParAccelRollback à la page 3078)
- PostgreSQL (tPostgresqlRollback à la page 3273)
- PostgresPlus (tPostgresPlusRollback à la page 3226)
- SAPHana (tSAPHanaRollback à la page 3677)
- SQLite (tSQLiteRollback à la page 3870)
- Sybase (ASE et IQ) (tSybaseRollback à la page 4073)
- Teradata (tTeradataRollback à la page 4129)
- VectorWise (tVectorWiseRollback à la page 4198)
- Vertica (tVerticaRollback à la page 4240)
tDBRow

Ce composant exécute une requête SQL spécifiée sur une base de données.
Ce composant fonctionne avec différentes bases de données.

Propriétés du tDBRow Standard

Ces propriétés sont utilisées pour configurer le composant tDBRow s'exécutant dans le framework de Jobs Standard.
Le composant tDBRow Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d'entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Access (tAccessRow à la page 97)
- Amazon Mysql (tAmazonMysqlRow à la page 201)
- Amazon Oracle (tAmazonOracleRow à la page 227)
- Amazon Redshift (tRedshiftRow à la page 3348)
- AS400 (tAS400Row à la page 264)
- Exasol (tEXARow à la page 967)
- FireBird (tFirebirdRow à la page 1262)
- Greenplum (tGreenplumRow à la page 1426)
- IBM DB2 (tDB2Row à la page 616)
- Informix (tInformixRow à la page 1845)
- Ingres (tIngresRow à la page 1892)
- Interbase (tInterbaseRow à la page 1918)
- JDBC (tJDBCRow à la page 1992)
- MemSQL (tMemSQLRow à la page 2374)
- Microsoft SQL Server (tMSSqlRow à la page 2579)
- MySQL (tMysqlRow à la page 2696)
- Netezza (tNetezzaRow à la page 2861)
- Oracle (tOracleRow à la page 2955)
- ParAccel (tParAccelRow à la page 3080)
• PostgreSQL (tPostgresqlRow à la page 3275)
• PostgresPlus (tPostgresPlusRow à la page 3228)
• SAPHana (tSAPHanaRow à la page 3679)
• SQLite (tSQLiteRow à la page 3872)
• Snowflake (tSnowflakeRow à la page 3773)
• Sybase (ASE et IQ) (tSybaseRow à la page 4075)
• Teradata (tTeradataRow à la page 4131)
• VectorWise (tVectorWiseRow à la page 4200)
• Vertica (tVerticaRow à la page 4242)
tDBSCD

Ce composant suit et reflète les modifications dans une base de données SCD spécifiée.
Ce composant fonctionne avec différentes bases de données.

Propriétés du tDBSCD Standard

Ces propriétés sont utilisées pour configurer le composant tDBSCD s’exécutant dans le framework de Jobs Standard.
Le composant tDBSCD Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Greenplum (tGreenplumSCD à la page 1431)
- IBM DB2 (tDB2SCD à la page 621)
- Informix (tInformixSCD à la page 1850)
- Ingres (tIngresSCD à la page 1896)
- Microsoft SQL Server (tMSSqlSCD à la page 2585)
- MySQL (tMysqlSCD à la page 2712)
- Netezza (tNetezzaSCD à la page 2866)
- Oracle (tOracleSCD à la page 2961)
- ParAccel (tParAccelSCD à la page 3085)
- PostgreSQL (tPostgresqlSCD à la page 3280)
- PostgresPlus (tPostgresPlusSCD à la page 3233)
- Sybase (ASE et IQ) (tSybaseSCD à la page 4080)
- Teradata (tTeradataSCD à la page 4137)
- Vertica (tVerticaSCD à la page 4247)
tDBSCDELT

Ce composant suit et reflète des modifications dans une table SCD dédiée via des requêtes SQL.
Ce composant fonctionne avec différentes bases de données.

Propriétés du tDBSCDELT Standard

Ces propriétés sont utilisées pour configurer le composant tDBSCDELT s’exécutant dans le framework de Jobs Standard.
Le composant tDBSCDELT Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

• IBM DB2 (tDB2SCDELT à la page 625)
• MySQL (tMysqlSCDELT à la page 2727)
• Oracle (tOracleSCDELT à la page 2966)
• PostgreSQL (tPostgresqlSCDELT à la page 3284)
• PostgresPlus (tPostgresPlusSCDELT à la page 3237)
• Sybase (ASE et IQ) (tSybaseSCDELT à la page 4084)
• Teradata (tTeradataSCDELT à la page 4141)
Ce composant appelle une procédure stockée de base de données.
Ce composant fonctionne avec différentes bases de données.

**Propriétés du tDBSP Standard**

Ces propriétés sont utilisées pour configurer le composant tDBSP s’exécutant dans le framework de JobsStandard.
Le composant tDBSP Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

**Basic settings**

Ce composant sert de point d’entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste Database et cliquez sur Apply dans sa vue Basic settings. Pour plus d’informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- IBM DB2 (tDB2SP à la page 630)
- Informix (tInformixSP à la page 1854)
- JDBC (tJDBCSP à la page 2010)
- Microsoft SQL Server (tMSSqlSP à la page 2590)
- MySQL (tMysqlSP à la page 2732)
- Oracle (tOracleSP à la page 2971)
- Sybase (ASE et IQ) (tSybaseSP à la page 4089)
tDBTableList

Ce composant liste les noms des tables de bases de données spécifiées à l'aide d'une instruction SELECT basée sur une clause WHERE.

Ce composant fonctionne avec différentes bases de données.

**Propriétés du tDBTableList Standard**

Ces propriétés sont utilisées pour configurer le composant tDBSP s'exécutant dans le framework de Jobs Standard.

Le composant tDBSP Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

Ce composant sert de point d'entrée aux bases de données suivantes. Pour configurer ce composant, sélectionnez un type de base de données dans la liste **Database** et cliquez sur **Apply** dans sa vue **Basic settings**. Pour plus d'informations concernant les propriétés spécifiques aux bases de données, consultez la documentation associée :

- Microsoft SQL Server (**tMSSqlTableList** à la page 2600)
- MySQL (**tMysqlTableList** à la page 2739)
- Oracle (**tOracleTableList** à la page 2980)
tDBSQLRow

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler), selon la nature de la requête et de la base de données. Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Le tDBSQLRow est le composant générique à tout type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie. Pour une question de performances, préférez toujours un composant de base de données spécifique au composant générique.

Pour utiliser ce composant, les pilotes ODBC du SGBD doivent être installés et les connexions ODBC doivent être configurées via l’assistant de configuration de connexion à une base de données.

Propriétés du tDBSQLRow Standard

Ces propriétés sont utilisées pour configurer le tDBSQLRow s’exécutant dans le framework de Jobs Standard.

Le composant tDBSQLRow Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Datasource</td>
<td>Nom de la source de données définie dans l’assistant de configuration de la connexion à la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Nom de la table de base de données à modifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez la requête appropriée dans le Repository. Le champ <em>Query</em> est renseigné automatiquement.</td>
</tr>
<tr>
<td>Query</td>
<td>Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur. Si nécessaire, vous pouvez récupérer les lignes en erreur via un lien <strong>Row &gt; Rejects</strong>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Additional JDBC parameters</th>
<th>Spécifiez les propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque</strong> :</td>
</tr>
<tr>
<td></td>
<td>Vous pouvez configurer les paramètres d’encodage dans ce champ.</td>
</tr>
<tr>
<td>Propagate QUERY’s recordset</td>
<td>Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste <strong>use column</strong>.</td>
</tr>
<tr>
<td>Use PreparedStatement</td>
<td>Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de</td>
</tr>
</tbody>
</table>
données. Dans le tableau **Set PreparedStatement**
**Parameter**, définissez les valeurs des paramètres
représentés par des `?` dans l'instruction SQL définie
dans le champ **Query** de l'onglet **Basic settings**.

**Parameter Index** : Saisissez la position du paramètre
dans l'instruction SQL.

**Parameter Type** : Saisissez le type du paramètre.

**Parameter Value** : Saisissez la valeur du paramètre.

**Remarque** :
Cette option est très utile si vous devez
effectuer de nombreuses fois la même requête.
Elle permet un gain de performance.

---

**Commit every**
Nombre de lignes à inclure dans le lot avant de
commencer l'écriture dans la base. Cette option garantit
la qualité de la transaction (cependant pas de rollback)
et surtout une meilleure performance d'exécution.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au
niveau du composant.

---

**Global Variables**

**GLOBAL VARIABLES**

**QUERY** : requête traitée. Cette variable est une variable
Flow et retourne une chaîne de caractères.

**ERROR MESSAGE** : message d'erreur généré par le
composant lorsqu’une erreur survient. Cette variable est
une variable After et retourne une chaîne de caractères.
Cette variable fonctionne uniquement si la case **Die on error**
est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution
d’un composant. Une variable After fonctionne après
l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide
d’une variable, appuyez sur les touches Ctrl+Espace
pour accéder à la liste des variables. A partir de cette
liste, vous pouvez choisir la variable que vous souhaitez
utiliser.

Pour plus d’informations concernant les variables,
consultez le Guide utilisateur du Studio Talend.

---

**Utilisation**

**Règle d’utilisation**
Ce composant offre la flexibilité des requêtes sur les
bases de données et couvre toutes les possibilités de
requêtes SQL.

Utilisez le composant DBRow correspondant au type de
base de données dont vous vous servez. La plupart des
bases de données ont leur propre composant DBRow
spécifique.
**Prérequis**

ODBC (Open DataBase Connectivity) est un protocole standard permettant d'accéder aux informations de serveurs de bases de données. Vous devez préalablement créer une source de données (datasource) ODBC et la relier à votre base de données pour en lire les informations.

Pour créer une source ODBC sous Windows, par exemple, allez à Panneau de configuration > Outils d'administration > Sources de données (ODBC) puis renseignez les propriétés de la source à partir de l'assistant.
**tDenormalize**

Ce composant dénormalise un flux entrant en fonction d’une colonne.

**Propriétés du tDenormalize Standard**

Ces propriétés sont utilisées pour configurer le tDenormalize s’exécutant dans le framework de Jobs Standard.

Le composant tDenormalize Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
<td></td>
</tr>
<tr>
<td>Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To denormalize</th>
<th>Ce tableau vous permet de choisir les paramètres de dénormalisation des colonnes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column : Sélectionnez la colonne à dénormaliser.</td>
<td></td>
</tr>
<tr>
<td>Delimiter : Saisissez, entre guillemets doubles, le séparateur que vous souhaitez utiliser.</td>
<td></td>
</tr>
<tr>
<td>Merge same value : Cochez cette case si vous voulez fusionner les valeurs identiques.</td>
<td></td>
</tr>
</tbody>
</table>
Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. Notez que cette case n’est pas disponible dans la version Map/Reduce de ce composant. |

Global Variables

| Global Variables | ERROR MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
| | NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. |
| | Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |
| | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
| | Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en étape intermédiaire dans un flux de données. |
| Limitation | Notez que ce composant peut changer l’ordre du flux Java entrant. |

Scénario : Dénormaliser une colonne

Ce scénario illustre un Job dénormalisant une colonne d’un fichier délimité via un composant de dénormalisation, et affichant le résultat dans la console.

Dénormaliser une colonne

Procédure

1. Cliquez et déposez les composants suivants : tFileInputDelimited, tDenormalize, tLogRow.
2. Connectez ces composants à l'aide de liens **Row main**.

3. Dans la vue **Component** du composant **tFileInputDelimited**, renseignez le chemin d'accès au fichier à dénormaliser.

4. Dans les champs **Header**, **Row Separator** et **Field Separator**, renseignez respectivement le nombre de lignes d'en-tête, le type de séparateur de lignes et de champs.

5. Le fichier d’entrée est composé de deux colonnes : *Fathers* et *Children*.

6. Dans l’onglet **Basic settings** du **tDenormalize**, paramétrez la colonne contenant des valeurs multiples à regrouper. Dans le tableau **To denormalize**, sélectionnez la colonne *Children*.

7. Dans cet exemple, la colonne *Children* est celle à dénormaliser.


9. Cochez la case **Merge same value** si vous souhaitez supprimer les éventuels doublons.

10. Enregistrez le Job et exécutez-le en appuyant sur **F6**.
Résultats
Starting job tDenormalize at 10:50 09/03/2016.

Mickael | Océane
Stéphane | Agathe | Clémence
Pierrick | Erwan | Tiphaine
Robert | Xenon
Richard | Héctor
Fabrice | Martin
Job tDenormalize ended at 10:50 09/03/2016. [exit code=0]

Toutes les valeurs de la colonne Children (définie comme colonne à dénormaliser) sont regroupées en fonction des valeurs de la colonne Fathers. Les valeurs sont séparées par des virgules.

Scénario 2 : Dénormaliser plusieurs colonnes

Ce scénario illustre un Job dénormalisant deux colonnes d’un fichier délimité via un composant de dénormalisation et affichant le résultat dans la console.

Scénario 2 : Dénormaliser plusieurs colonnes

Procédure
1. Cliquez et déposez les composants suivants : tFileInputDelimited, tDenormalize, tLogRow.
2. Connectez tous les composants à l’aide d’une connexion Row > main.
3. Dans l’onglet Basic settings du composant tFileInputDelimited, renseignez le chemin d’accès au fichier à dénormaliser.

4. Renseignez les autres champs, dont les champs Row Separator, Field separators et Header contenant respectivement les séparateurs de lignes et de champs et le nombre de lignes d’en-tête.
5. Le schéma du fichier est composé de quatre colonnes : Name, FirstName, HomeCity, WorkCity.
6. Dans l’onglet Basic settings du composant tDenormalize, sélectionnez les colonnes contenant les répétitions. Elles correspondent aux colonnes apparaissant plusieurs fois dans le document. Dans cet exemple, FirstName, HomeCity et WorkCity sont les colonnes à partir desquelles la dénormalisation est effectuée.

7. Ajoutez autant trois lignes à la table To denormalize que vous le souhaitez en cliquant sur le bouton [+]. Puis sélectionnez les colonnes FirstName, HomeCity et WorkCity dans la liste déroulante.

8. Dans la colonne Delimiter, paramétrez les séparateurs en les saisissant entre guillemets, afin de séparer les valeurs concaténées. Pour la colonne FirstName, saisissez “#”, pour la colonne HomeCity, saisissez “§”, et pour la colonne WorkCity, saisissez “¤”.


Le résultat affiche les valeurs dénormalisées concaténées à l’aide des séparateurs.

10. Dans l’onglet Basic settings du composant tDenormalize, dans le tableau To denormalize, cochez la case Merge same value pour fusionner les valeurs identiques.

Résultats

Cette fois-ci, la console affiche les résultats sans doublon.
tDenormalizeSortedRow

Ce composant permet de synthétiser les lignes d’entrée triées afin de faire des économies de mémoire.

Le tDenormalizeSortedRow rassemble dans un groupe toutes les lignes d’entrée dénormalisées triées ainsi que leurs valeurs distinctes, associées par des séparateurs de champ.

Propriétés du tDenormalizeSortedRow Standard

Ces propriétés sont utilisées pour configurer le tDenormalizeSortedRow s’exécutant dans le framework de Jobs Standard.

Le composant tDenormalizeSortedRow Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Sync columns pour récupérer le schéma du composant précédent.</td>
</tr>
<tr>
<td>Input rows count</td>
<td>Renseignez le nombre de lignes d’entrée.</td>
</tr>
</tbody>
</table>


To denormalize

Renseignez le nom de la colonne à dénormaliser.

Advanced settings

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

Global Variables

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant traite des flux de données, il nécessite donc un composant d’entrée et un composant de sortie.

Scénario : Regrouper des lignes triées

Ce scénario Java décrit un Job à quatre composants. Il permet de lire un fichier délimité donné ligne par ligne, de trier les données d’entrées en fonction de leur type et de leur ordre, de dénormaliser toutes les lignes d’entrées triées et enfin d’afficher le résultat dans la console de la vue Run.

• A partir de la Palette, cliquez-déposez les composants suivants dans l’éditeur graphique : le tFileInputDelimited, le tSortRow, le tDenormalizeSortedRow et le tLogRow.
• Connectez les quatre composants à l’aide de liens de type Row Main.
• Dans l’éditeur graphique, sélectionnez le composant `tFileInputDelimited`.

• Cliquez sur la vue `Component` pour définir la configuration de base (Basic settings) du `tFileInputDelimited`.

![tFileInputDelimited_1](image)

<table>
<thead>
<tr>
<th>Basic settings</th>
<th>Property Type: Built-In</th>
<th>File Name: &quot;C:/tests/name_list.txt&quot;</th>
<th>Row Separator: ;</th>
<th>Field Separator:</th>
<th>CSV options:</th>
<th>Header: 0</th>
<th>Footer: 0</th>
<th>Limit:</th>
<th>Schema: Built-In</th>
<th>Edit schema:</th>
</tr>
</thead>
</table>

• Définissez le champ `Property Type` en mode `Built-In`.

• Renseignez le chemin d’accès au fichier à traiter dans le champ `File Name`. Dans cet exemple, on utilise le fichier `name_list`, qui comporte deux colonnes, `id` et `first name`.

• Définissez si nécessaire les séparateurs de lignes et de champs, l’en-tête et le pied-de-page, ainsi que le nombre de lignes traitées.

• Définissez le champ `Schema` en mode `Built in` puis cliquez sur le bouton ` [...]` à côté du champ `Edit Schema` pour définir les données à passer au composant suivant. Dans cet exemple, le schéma est constitué de deux colonnes, `id` et `name`. 
• Dans l'espace graphique, sélectionnez le composant tSortRow.
• Cliquez sur la vue Component pour en définir la configuration de base (Basic settings) du tSortRow.

  ![tSortRow_1](image)

• Définissez le champ Schema Type en mode Built-In puis cliquez sur Sync columns pour récupérer le schéma à partir du composant tFileInputDelimited.
• Dans le panneau Criteria, cliquez sur le bouton [+] pour ajouter une ligne puis définissez les paramètres de tri pour la colonne du schéma à traiter. Dans cet exemple, on veut trier la colonne id dans l'ordre croissant.
• Dans l'espace graphique, sélectionnez le composant tDenormalizeSortedRow.
• Cliquez sur la vue Component pour définir la configuration de base (Basic settings) du tDenormalizeSortedRow.

  ![tDenormalizeSortedRow_1](image)

Warning: This component may change input behavior.
• Définissez le champ Schema en mode Built-In puis cliquez sur Sync columns pour récupérer le schéma à partir du schéma du composant tSortRow.

• Dans le champ Input rows count, saisissez le nombre de lignes d’entrée à traiter ou cliquez simultanément sur Ctrl+Espace pour accéder à la liste des variables de contexte puis sélectionnez la variable : tFileInputDelimited_1_NB_LINE

• Dans le panneau To denormalize, cliquez sur le bouton [+] pour ajouter une ligne puis définissez les paramètres de la colonne à dénormaliser. Dans cet exemple, on veut dénormaliser la colonne name.

• Dans l’éditeur graphique, sélectionnez le composant tLogRow puis cliquez sur la vue Component pour en définir la configuration de base (Basic settings). Pour plus d’informations sur le tLogRow, consultez tLogRow à la page 2105.

• Enregistrez votre Job puis appuyez sur F6 pour l’exécuter.

Le résultat affiché sur la console montre la façon dont la colonne name a été dénormalisée.
tDie

Ce composant déclenche le composant tLogCatcher afin de collecter des données de log exhaustives avant de tuer le Job.

Les composants tDie et tWarn sont étroitement liés au composant tLogCatcher. Ils sont généralement utilisés ensemble afin que les données de log collectées par le tLogCatcher soient rassemblées et envoyées vers la sortie définie.

Le composant tDie envoie une erreur et arrête le Job en cours d'exécution. Si vous souhaitez uniquement envoyer un message d’avertissement, consultez tWarn à la page 4271.

Propriétés du tDie Standard

Ces propriétés sont utilisées pour configurer le tDie s’exécutant dans le framework de Jobs Standard. Le composant tDie Standard appartient à la famille Logs & Errors. Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Die message</th>
<th>Saisissez le message qui sera affiché avant que le Job ne soit tué.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error code</td>
<td>Saisissez le code de l'erreur (un entier) si nécessaire.</td>
</tr>
<tr>
<td>Priority</td>
<td>Sélectionnez le niveau de priorité.</td>
</tr>
</tbody>
</table>

Remarque :
Notez que toute valeur supérieure à 255 ne peut pas être utilisée comme code d’erreur sous Linux.

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>DIE_MESSAGES : message d’arrêt. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DIE_CODE : code d’erreur du message d’arrêt. Cette variable est une variable After et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td>DIE_PRIORITY : niveau de priorité du message d’arrêt. Cette variable est une variable After et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Scénarios associés

Pour des exemples d’utilisation du composant **tDie**, consultez les scénarios du composant **tLogCatcher** :

- **Capturer les messages déclenchés par un composant tWarn** à la page 2098.
- **Capturer le message déclenché par un composant tDie** à la page 2101.
**tDotNETInstantiate**

Ce composant invoque le constructeur d'un objet .NET pour un usage ultérieur.
Le tDotNETInstantiate instancie un objet dans .NET pour une utilisation ultérieure.

**Propriétés du tDotNETInstantiate Standard**

Ces propriétés sont utilisées pour configurer le tDotNETInstantiate s'exécutant dans le framework de Jobs Standard.
Le composant tDotNETInstantiate Standard appartient à la famille DotNET.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dll to load</td>
<td>Saisissez dans ce champ le chemin d'accès à une bibliothèque DLL contenant la (les) classe(s) qui vous intéresse(nt) ou saisissez le nom de l'Assembly ou cliquez sur le bouton [...] afin de parcourir votre système jusqu'au répertoire de votre bibliothèque. Par exemple, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089 pour une Assembly OleDb.</td>
</tr>
<tr>
<td>Fully qualified class name(i.e. ClassLibrary1. NameSpace2.Class1)</td>
<td>Renseignez ce champ en saisissant le chemin complet de la classe (FQN).</td>
</tr>
<tr>
<td>Value(s) to pass to the constructor</td>
<td>Cliquez sur le bouton [+] pour ajouter une ou plusieurs valeur(s) à passer au constructeur de l'objet. Ou, laissez vide ce tableau afin d'appeler un constructeur par défaut pour l'objet. La/les valeur(s) valide(s) doit(ent)t être les paramètres requis par la classe à utiliser.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Variables</td>
<td>INSTANCE : instance d'un objet .NET. Cette variable est une variable After et retourne un object.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.</td>
</tr>
</tbody>
</table>
Pour renseigner un champ ou une expression à l'aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

#### Règle d’utilisation

Ce composant est utilisé en tant que composant de début dans un flux ou dans un sous-job indépendant.

Pour utiliser ce composant, vous devez d’abord installer les fichiers DLLs d’exécution, par exemple *janet-win32.dll* sous Windows 32 bits et *janet-win64.dll* sous Windows 64 bits, dans le package redistribuable Microsoft Visual C++, afin d’éviter les erreurs telles que *UnsatisfiedLinkError* sur les DLL dépendants.

Assurez-vous que le DLL d’exécution et tous les autres DLL dont dépend le DLL à appeler sont entièrement installés et que leurs versions sont en adéquation.

**Remarque :**

Les DLL d’exécution requis peuvent être installés dans le dossier *System32* ou dans le dossier *bin* du JRE à utiliser.

Si vous devez exporter un Job utilisant ce composant pour l’exécuter hors du *Studio Talend*, vous devez spécifier le conteneur d’exécution souhaité en configurant l’argument *-Djava.library.path*.

Les utilisateurs des solutions *Talend* avec ESB doivent copier les fichiers DLL du Runtime dans le répertoire *%KARAF_HOME%/lib/wrapper/* pour exécuter un Job contenant ce composant dans le Runtime ESB.

### Scénario associé

Pour un scénario associé, consultez Scénario : Utiliser .NET dans Talend à la page 680 du composant tDotNETRow.
**tDotNETRow**

Ce composant facilite la transformation de données à l’aide de classes .NET built-in ou personnalisées.
Le tDotNETRow envoie des données à partir de bibliothèques et vers des bibliothèques et des classes dans .NET ou d’autres fichiers DLL personnalisés.

**Propriétés du tDotNETRow Standard**

Ces propriétés sont utilisées pour configurer le tDotNETRow s’exécutant dans le framework de Jobs Standard.
Le composant tDotNETRow Standard appartient à la famille DotNET.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Schema et Edit schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant (Repository).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use a static method</td>
<td>Cochez cette case pour invoquer une méthode statique dans .NET. Cela désactivera la case <strong>Use an existing instance</strong>.</td>
</tr>
<tr>
<td>Propagate a data to output</td>
<td>Cochez cette case afin de propager des données transformées vers la sortie</td>
</tr>
<tr>
<td>Use an existing instance</td>
<td>Cochez cette case pour réutiliser une instance existante d’un objet .NET dans la liste <strong>Existing instance to use</strong>.</td>
</tr>
<tr>
<td><strong>Existing instance to use</strong></td>
<td>Sélectionnez une instance existante d’objets .NET créés par d’autres composants .NET de la liste.</td>
</tr>
</tbody>
</table>

**Remarque :**
Cette case est désactivée si vous avez coché **Use a static method**. Cochez cette case désactivera les champs **Dll to load**, **Fully qualified class name** (i.e. ClassLibrary1. NameSpace2.Class1) et **Value(s) to pass to the constructor**.

| Dll to load | Saisissez dans ce champ le chemin d’accès à une bibliothèque DLL contenant la (les) classe(s) qui |
vous intéresse(nt) ou saisissez le nom de l’Assembly ou cliquez sur le bouton […] afin de parcourir votre système jusqu’au répertoire de votre bibliothèque. Par exemple, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089 pour une Assembly OleDb.

<table>
<thead>
<tr>
<th>Fully qualified class name(i.e. ClassLibrary1.NameSpace2.Class1)</th>
<th>Renseignez ce champ en saisissant le chemin complet de la classe (FQN).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method name</td>
<td>Renseignez ce champ en saisissant le nom de la méthode à invoquer dans .NET.</td>
</tr>
<tr>
<td>Value(s) to pass to the constructor</td>
<td>Cliquez sur le bouton [+] pour ajouter une ou plusieurs valeur(s) à passer au constructeur de l’objet. Ou, laissez vide ce tableau afin d’appeler un constructeur par défaut pour l’objet. La/les valeur(s) valide(s) doi(ven)t être les paramètres requis par la classe à utiliser.</td>
</tr>
<tr>
<td>Method Parameters</td>
<td>Cliquez sur le bouton [+] pour ajouter une ou plusieurs ligne(s) de paramètres à passer à la méthode.</td>
</tr>
<tr>
<td>Output value target column</td>
<td>Dans la liste, sélectionnez une colonne de la lignes de sortie afin de lui attribuer une valeur.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Create a new instance at each row</th>
<th>Cochez cette case pour créer une nouvelle instance à chaque ligne qui passe dans le composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method doesn’t return a value</td>
<td>Cochez cette case afin d’invoquer une méthode sans retourner de valeur comme résultat du traitement.</td>
</tr>
<tr>
<td>Returns an instance of a .NET Object</td>
<td>Cochez cette case pour retourner une instance d’un objet .NET comme résultat d’une méthode invoquée.</td>
</tr>
<tr>
<td>Store the returned value for later use</td>
<td>Cochez cette case afin de stocker la valeur retournée d’une méthode pour une utilisation ultérieure dans un autre composant tDotNETRow.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| Global Variables                                           | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espacement** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

---

### Utilisation

| Règle d’utilisation | Ce composant est utilisé pour communiquer avec des objets .NET.

Pour utiliser ce composant, vous devez d’abord installer les fichiers DLLs d’exécution, par exemple `janet-win32.dll` sous Windows 32 bits et `janet-win64.dll` sous Windows 64 bits, dans le package redistribuable Microsoft Visual C++, afin d’éviter les erreurs telles que *UnsatisfiedLinkError* sur les DLL dépendants.

Assurez-vous que le DLL d’exécution et tous les autres DLL dont dépend le DLL à appeler sont entièrement installés et que leurs versions sont en adéquation.

**Remarque :**

Les DLLs d’exécution requis peuvent être installés dans le dossier *System32* ou dans le dossier *bin* du JRE à utiliser.

Si vous devez exporter un Job utilisant ce composant pour l’ exécuter hors du **Studio Talend**, vous devez spécifier le conteneur d’exécution souhaité en configurant l’ argument `-Djava.library.path`.

Les utilisateurs des solutions **Talend** avec ESB doivent copier les fichiers DLL du Runtime dans le répertoire `%KARAF_HOME%/lib/wrapper/` pour exécuter un Job contenant ce composant dans le Runtime ESB.

---

### Scénario : Utiliser .NET dans Talend

Ce scénario décrit un Job à trois composants utilisant une bibliothèque DLL contenant une classe appelée `Test1.Class1`. On invoque une méthode de cette classe qui traite la valeur et écrit les données dans la console.
Prérequis

Avant de reproduire ce scénario, vous devez d’abord construire votre environnement d’exécution.

• Créez le DLL à charger par le tDotNETInstantiate

Cette classe d’exemple construite dans .NET se présente comme suit :

```
using System;
using System.Collections.Generic;
using System.Text;
namespace Test1
{
    public class Class1
    {
        string s = null;
        public Class1(string s)
        {
            this.s = s;
        }
        public string getValue()
        {
            return "Return Value from Class1: " + s;
        }
    }
}
```

Cette classe lit la valeur d’entrée et ajoute le texte Return Value from Class1: devant cette valeur. La classe est compilée à l’aide du dernier .NET.

• Installez le fichier DLL d’exécution à partir du dernier .NET. Dans ce scénario, utilisez janet-win32.dll sous Windows 32 bits et placez-le dans le dossier System32.

Le DLL d’exécution est compatible avec le DLL à charger.

Relier les composants

Procédure

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : tDotNETInstantiate, tDotNETRow et tLogRow.
2. Connectez le tDotNETInstantiate au tDotNETRow à l’aide d’un lien Trigger > OnSubjobOk.
3. Reliez le tDotNETRow au tLogRow à l'aide d'un lien Row > Main.

**Configurer le tDotNETInstantiate**

**Procédure**

1. Double-cliquez sur le tDotNETInstantiate pour afficher sa vue Basic settings et définir ses propriétés de base.

2. Cliquez sur le bouton [...] à côté du champ Dll to load et parcourez votre système jusqu'au fichier dll à charger. Vous pouvez également renseigner le champ avec une Assembly. Dans cet exemple, utilisez :
   
   "C:/Program Files/ClassLibrary1/bin/Debug/ClassLibrary1.dll"

3. Dans le champ Fully qualified class name, saisissez un nom de classe valide à utiliser. Dans cet exemple, saisissez :
   
   "Test1.Class1"

4. Cliquez sur le bouton [+ ] sous la table Value(s) to pass to the constructor afin d’ajouter une ligne pour la valeur à passer au constructeur.
   
   Dans cet exemple, saisissez :
   
   "Hello world"

**Configurer le composant tDotNETRow**

**Procédure**

1. Double-cliquez sur le composant tDotNETRow afin d’afficher sa vue Basic settings et définir ses propriétés de base.
2. Cochez la case **Propagate data to output**.
3. Cochez la case **Use an existing instance** et sélectionnez **tDotNETInstantiate_1** dans la liste **Existing instance to use**.
4. Renseignez le champ **Method Name** avec le nom de la méthode à utiliser. Dans cet exemple, utilisez "getValue".
5. Cliquez sur le bouton [..] à côté du champ **Edit schema** pour ajouter une colonne au schéma.

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Type</th>
<th>N. Date Part...</th>
<th>Length</th>
<th>Pre...</th>
<th>D...</th>
<th>Co...</th>
</tr>
</thead>
<tbody>
<tr>
<td>newColumn</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cliquez sur le bouton [+] afin d’ajouter une colonne au schéma et cliquez sur **OK** pour sauvegarder les modifications.
6. Sélectionnez **newColumn** dans la liste **Output value target column**.
7. Laissez les autres paramètres tels qu’ils sont.

**Configurer le tLogRow**

**Procédure**

1. Double-cliquez sur le **tLogRow** afin d’afficher sa vue **Basic settings** et définir ses propriétés de base.
2. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma défini dans le composant précédent.

3. Sélectionnez **Table** dans la zone **Mode**.

**Résultats**

Sauvegardez votre Job et appuyez sur la touche **F6** pour l'exécuter.

![Execution screen](image)

Vous pouvez constater, dans les résultats, que le texte **Return Value from Class1** est ajouté devant la valeur récupérée **Hello world**.
tDropboxConnection

Ce composant crée une connexion Dropbox vers un compte donné, que les autres composants Dropbox peuvent réutiliser.

Propriétés du tDropboxConnection Standard

Ces propriétés sont utilisées pour configurer le tDropboxConnection s’exécutant dans le framework de Jobs Standard.

Le composant tDropboxConnection Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**


Notez qu’une application Dropbox doit avoir été créée au préalable sous ce compte avant que le jeton d’accès puisse être généré. Pour plus d’informations concernant les jetons d’accès Dropbox, consultez [https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account](https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account) (en anglais).

| Use HTTP Proxy | Si vous utilisez un proxy, cochez cette case et saisissez les informations d’hôte et de port de ce proxy dans les champs correspondants qui s’affichent.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Global Variables**

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

**Utilisation**

| Règle d’utilisation | Ce composant est utilisé en standalone en tant que sous-job pour créer une connexion Dropbox. Dans un Job, ce composant est généralement connecté aux autres composants Dropbox à l’aide de liens de type Trigger, tel que le lien On Subjob Ok. |
Scénario associé

Pour un scénario associé, consultez Scénario : Télécharger des fichiers dans Dropbox à la page 695
## tDropboxDelete

Ce composant supprime un dossier ou un fichier donné de Dropbox.

### Propriétés du tDropboxDelete Standard

Ces propriétés sont utilisées pour configurer le tDropboxDelete s’exécutant dans le framework de Jobs Standard.

Le composant tDropboxDelete Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

#### Basic settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Existing Connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td></td>
<td>Notez qu’une application Dropbox doit avoir été créée au préalable sous ce compte avant que le jeton d’accès puisse être généré. Pour plus d’informations concernant les jetons d’accès Dropbox, consultez <a href="https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account">https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account</a> (en anglais).</td>
</tr>
<tr>
<td><strong>Use HTTP Proxy</strong></td>
<td>Si vous utilisez un proxy, cochez cette case et saisissez les informations d’hôte et de port de ce proxy dans les champs correspondants qui s’affichent.</td>
</tr>
<tr>
<td><strong>Path</strong></td>
<td>Renseignez le chemin Dropbox d’accès au dossier ou fichier que vous souhaitez supprimer.</td>
</tr>
<tr>
<td></td>
<td>Notez que le nom du chemin d’accès doit commencer par une barre oblique (/). Il s’agit du dossier racine de l’application Dropbox pour laquelle vous utilisez le jeton d’accès.</td>
</tr>
</tbody>
</table>

#### Advanced settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>
Global Variables

| **ERROR_MESSAGE** | Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

Utilisation

| Règle d'utilisation | Ce composant est utilisé en standalone dans un sous-job pour supprimer des données de Dropbox. |

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tDropboxGet**

Ce composant télécharge un fichier sélectionné d’un compte Dropbox vers un répertoire local spécifié

**Propriétés du tDropboxGet Standard**

Ces propriétés sont utilisées pour configurer le tDropboxGet s’exécutant dans le framework de Jobs Standard.

Le composant tDropboxGet Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Use Existing Connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
</table>
Notez qu’une application Dropbox doit avoir été créée au préalable sous ce compte avant que le jeton d’accès puisse être généré. Pour plus d’informations concernant les jetons d’accès Dropbox, consultez https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account (en anglais). |
| Use HTTP Proxy          | Si vous utilisez un proxy, cochez cette case et saisissez les informations d’hôte et de port de ce proxy dans les champs correspondants qui s’affichent. |
| Path                    | Renseignez le chemin Dropbox d’accès au fichier que vous souhaitez télécharger. 
Notez que le nom du chemin d’accès doit commencer par une barre oblique (/). Il s’agit du dossier racine de l’application Dropbox pour laquelle vous utilisez le jeton d’accès. |
| Save As File            | Cochez cette case pour afficher le champ File et par courez votre système ou saisissez le chemin d’accès au répertoire local dans lequel vous souhaitez que soit stocké le fichier téléchargé. Le fichier est remplacé s’il existe déjà. |
| Schema et Edit schema   | Un schéma est une description de lignes. Il définit le nombre de champs (colonnes) à traiter et à passer au composant suivant. 
Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton à côté du champ Edit. |
### tDropboxGet

**Schema** pour voir le schéma prédéfini contenant les deux colonnes suivantes :
- **fileName** : nom du fichier téléchargé.
- **content** : contenu du fichier téléchargé.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. |

### Global Variables

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

### Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en standalone ou avec d’autres composants à l’aide d’un lien de type Iterate ou d’un lien de type Trigger tel que OnSubjobOK. |

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tDropboxList

Ce composant liste les fichiers stockés dans un répertoire spécifié dans Dropbox.

Propriétés du tDropboxList Standard

Ces propriétés sont utilisées pour configurer le tDropboxList s’exécutant dans le framework de Jobs Standard.

Le composant tDropboxList Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use Existing Connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use HTTP Proxy</td>
<td>Si vous utilisez un proxy, cochez cette case et saisissez les informations d’hôte et de port de ce proxy dans les champs correspondants qui s’affichent.</td>
</tr>
<tr>
<td>Path</td>
<td>Renseignez le chemin d’accès vers le dossier dont vous souhaitez lister les fichiers, ou renseignez le chemin d’accès vers le fichier exact que vous souhaitez lire. Notez que le nom du chemin d’accès doit commencer par une barre oblique (/). Il s’agit du dossier racine de l’application Dropbox pour laquelle vous utilisez le jeton d’accès.</td>
</tr>
<tr>
<td>List Type</td>
<td>Sélectionnez, dans le chemin d’accès spécifié, le type de données que vous souhaitez lister.</td>
</tr>
<tr>
<td>Include subdirectories</td>
<td>Cochez cette case pour lister les fichiers de n’importe quel sous-dossier existant en plus des fichiers dans le répertoire défini dans le champ Path.</td>
</tr>
</tbody>
</table>
Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

Global Variables

<table>
<thead>
<tr>
<th>NAME</th>
<th>Nom du fichier distant en cours de traitement. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH</td>
<td>Chemin du dossier ou du fichier en cours de traitement dans Dropbox. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>LAST_MODIFIED</td>
<td>Horodatage de la dernière modification du fichier traité. Cette variable est une variable Flow et retourne un long.</td>
</tr>
<tr>
<td>SIZE</td>
<td>Volume du fichier en cours de traitement. Cette variable est une variable Flow et retourne un long.</td>
</tr>
<tr>
<td>IS_FILE</td>
<td>Le booléen résultant de la liste des fichiers. Cette variable est une variable Flow et retourne un booléen. Le résultat <em>Yes</em> indique que les données listées sont de type File, sinon, leur type est Folder.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé en standalone. |

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tDropboxPut

Ce composant télécharge des données dans Dropbox à partir d'un fichier local ou d'un flux de données spécifié.

Propriétés du tDropboxPut Standard

Ces propriétés sont utilisées pour configurer le tDropboxPut s'exécutant dans le framework de Jobs Standard.

Le composant tDropboxPut Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Use Existing Connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access Token</strong></td>
<td>Renseignez le jeton d'accès requis par le compte Dropbox auquel vous souhaitez vous connecter. Ce jeton d'accès permet au Studio d'appeler l'API Dropbox pour ce compte Dropbox. Notez qu'une application Dropbox doit avoir été créée au préalable sous ce compte avant que le jeton d'accès puisse être généré. Pour plus d'informations concernant les jetons d'accès Dropbox, consultez <a href="https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account">https://www.dropbox.com/developers/blog/94/generate-an-access-token-for-your-own-account</a> (en anglais).</td>
</tr>
<tr>
<td><strong>Use HTTP Proxy</strong></td>
<td>Si vous utilisez un proxy, cochez cette case et saisissez les informations d'hôte et de port de ce proxy dans les champs correspondants qui s'affichent.</td>
</tr>
<tr>
<td><strong>Path (File Only)</strong></td>
<td>Renseignez le chemin d'accès vers le fichier dans lequel vous souhaitez écrire du contenu. Ce fichier est créé s'il n'existe pas. Notez que le nom du chemin d'accès doit commencer par une barre oblique (/). Il s'agit du dossier racine de l'application Dropbox pour laquelle vous utilisez le jeton d'accès.</td>
</tr>
<tr>
<td><strong>Upload Mode</strong></td>
<td>Sélectionnez le mode de téléchargement à utiliser :</td>
</tr>
<tr>
<td></td>
<td>• <strong>Rename if Existing</strong> : le fichier téléchargé dans Dropbox est automatiquement renommé. Par exemple, un fichier nommé test.txt peut être renommé en test (1).txt.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Replace if Existing</strong> : le fichier téléchargé dans Dropbox remplace le fichier déjà existant.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update specified Revision</strong> : le fichier que vous téléchargez dans Dropbox est utilisé pour mettre</td>
</tr>
</tbody>
</table>
à jour une révision spécifique de ce fichier. Si la révision que vous spécifiez est la dernière révision, alors le fichier existant dans Dropbox est remplacé. S'il s'agit une révision antérieure, le fichier que vous téléchargez est renommé pour indiquer qu'il y a un conflit. Si la révision n'existe pas, une erreur est retournée.

**Upload Incoming content as File**

 Sélectionnez ce bouton radio pour lire les données directement du flux d'entrée du composant précédent et écrire les données dans le fichier spécifié dans le champ Path.

**Schema et Edit schema**

 Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

 Notez que le schéma de ce composant est en lecture seule et est composé d’une seule colonne appelée content. Il reçoit des données de la colonne content de son schéma d’entrée uniquement. Cela signifie que vous devez utiliser une colonne content dans le flux de données d’entrée pour transporter les données à télécharger. Ce type de colonne est généralement fourni par le composant tFileInputRaw. Pour plus d’informations, consultez tFileInputRaw à la page 1139.

 Le champ Schema n’est pas disponible lorsque vous sélectionnez le bouton radio Expose as OutputStream ou Upload local file.

**Upload local file**

 Sélectionnez ce bouton radio pour télécharger dans Dropbox un fichier stocké localement. Dans le champ File qui s’affiche, vous devez saisir le chemin d'accès ou parcourir votre système jusqu’au fichier.

**Expose as OutputStream**

 Cochez cette case pour exposer le flux de sortie de ce composant en tant que variable appelée OUTPUTSTREAM, afin que les autres composants puissent réutiliser cette variable pour écrire le contenu à télécharger dans le flux de sortie exposé.

 Par exemple, vous pouvez utiliser la fonctionnalité Use output stream du composant tFileOutputDelimited pour alimenter le flux de sortie exposé d’un tDropboxPut donné. Pour plus d’informations, consultez tFileOutputDelimited à la page 1169.

**Advanced settings**

**tStatCatcher Statistics**

 Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.
Global Variables

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

Utilisation

| Règle d’utilisation | Ce composant est soit utilisé en standalone dans un sous-job pour télécharger un fichier local directement dans Dropbox, soit utilisé en tant que composant de sortie d’un flux de Job pour télécharger des données manipulées dans ce flux. |

Scénario : Télécharger des fichiers dans Dropbox

Dans ce scénario, un Job à six composants composé de trois sous-jobs est créé pour écrire des données dans Dropbox à l’aide de différents modes de téléchargement.

Avant de reproduire ce scénario, vous devez créer une application Dropbox sous le compte Dropbox utilisé. Dans ce scénario, l’application Dropbox utilisée est appelée talendrop. Le fichier racine dans lequel les fichiers sont téléchargés est appelé talenddrop également. En outre, le jeton d’accès à ce fichier a été généré à partir de la console de l’application fournie par Dropbox.


Relier les composants

Procédure

1. Dans la perspective Integration du Studio, créez un Job vide à partir du nœud Job Designs dans la vue Repository.
Pour plus d’informations concernant la création d’un Job, consultez le Guide utilisateur du Studio Talend.

2. Dans l’espace de modélisation, saisissez le nom du composant à utiliser et sélectionnez ce composant dans la liste qui apparaît. Dans ce scénario, les composants utilisés sont le tDropboxConnection, le tFixedFlowInput, le tFileOutputDelimited, le tFileInputRaw et deux composants tDropboxPut.

Dans ce scénario, le composant tFixedFlowInput génère des données à télécharger dans Dropbox. Dans des cas réels, vous pouvez utiliser d’autres composants tels que le tMysqlInput ou le tMap à la place du tFixedFlowInput, afin de concevoir un processus plus élaboré pour préparer le traitement de vos données.

3. Reliez le tFixedFlowInput au tFileOutputDelimited à l’aide d’un lien de type Row > Main.
4. Faites de même pour relier le tFileOutputDelimited à l’un des deux composants tDropboxPut et reliez le tFileInputRaw à l’autre composant tDropboxPut.
5. Reliez le tDropboxConnection au tFixedFlowInput à l’aide d’un lien de type Trigger > OnSubjobOk, puis reliez le tFixedFlowInput au tFileInputRaw à l’aide d’un lien de même type.

Se connecter à Dropbox

Procédure

1. Double-cliquez sur le tDropboxConnection pour ouvrir sa vue Component.

2. Dans le champ Access token, collez le jeton que vous avez généré via la console de l’application de Dropbox afin d’accéder au dossier de l’application Dropbox à utiliser.

Générer le flux de sortie

Définir les données d’entrée

Procédure

1. Double-cliquez sur le tFixedFlowInput pour ouvrir sa vue Component.
Dans ce scénario, seules trois lignes de données sont créées pour indiquer trois pays et leurs indicatifs téléphoniques.

<table>
<thead>
<tr>
<th>Code</th>
<th>Pays</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>France</td>
</tr>
<tr>
<td>86</td>
<td>Chine</td>
</tr>
<tr>
<td>81</td>
<td>Japon</td>
</tr>
</tbody>
</table>

2. Cliquez sur le bouton [...] à côté de Edit schema pour ouvrir l’éditeur de schéma.
3. Cliquez deux fois sur le bouton [+] pour ajouter deux lignes et, dans la colonne Column, renommez-les code et country.

4. Cliquez sur OK pour valider ces changements et accepter la propagation demandée par la boîte de dialogue qui s’ouvre.
5. Dans la zone Mode, sélectionnez le bouton radio Use Inline Table. Les colonnes code et country sont automatiquement créées dans cette table.
Définir le flux de sortie

Procédure

1. Double-cliquez sur le tFileOutputDelimited pour ouvrir sa vue Component.

2. Cochez la case **Use output stream** pour écrire les données dans un flux de sortie donné.

3. Dans le champ **Output stream**, renseignez le code pour définir le flux de sortie dans lequel vous souhaitez écrire les données. Dans ce scénario, il s’agit du flux de sortie du composant **tDropboxPut_1** relié au composant actuel. Le code utilisé pour écrire les données se présente de la manière suivante : 
```java
((java.io.OutputStream)globalMap.get("tDropboxPut_1_OUTPUTSTREAM"))
```

Notez que dans cet exemple de code, le composant **tDropboxPut** a le chiffre 1 comme affixe. Il correspond à son ID de composant automatiquement attribué dans ce Job. Si le composant **tDropboxPut** que vous utilisez a un ID différent, vous devez adapter le code en fonction de cet ID.

4. Cliquez sur **Edit schema** pour vérifier que le schéma de ce composant est identique à celui du composant **tFixedFlowInput** qui précède. Si ce n’est pas le cas, cliquez sur le bouton **Sync columns** pour rendre les deux schémas identiques.

Exposer le flux de sortie du tDropboxPut

Procédure

1. Double-cliquez sur le composant **tDropboxPut** relié au tFileOutputDelimited pour ouvrir sa vue Component.
2. Cochez la case **Use existing connection** pour réutiliser la connexion créée par le tDropboxConnection.

3. Dans le champ **Path**, saisissez le chemin d’accès au fichier dans lequel vous souhaitez écrire les données en le faisant précéder d’une barre oblique (/). Par exemple, saisissez */calling_code.csv*.

4. Dans la zone **Upload mode**, sélectionnez le bouton radio **Rename if Existing**.

5. Sélectionnez le bouton radio **Expose As OutputStream** pour exposer le flux de sortie de ce composant afin que l’autre composant, tFileOutputDelimited dans ce scénario, puisse écrire des données dans le flux.

### Définir les données du média à télécharger

**Procédure**

1. Double-cliquez sur le tFileInputRaw pour ouvrir sa vue **Component**.

   ![Component](image)

   Ce composant est utilisé pour lire une image appelée *esb_architecture.png* dans le flux de données. Dans des cas réels, ce fichier peut avoir de nombreux autres formats tels que PDF, XLS, PPT ou MP3.

2. Dans le champ **Filename**, saisissez le chemin d’accès ou parcourez votre système jusqu’au fichier que vous souhaitez télécharger.

3. Dans la zone **Mode**, sélectionnez le bouton radio **Read the file as a bytes array**.

### Mettre à jour le contenu entrant

**Procédure**

1. Double-cliquez sur le composant tDropboxPut relié au tFileInputRaw pour ouvrir sa vue **Component**.
2. Cochez la case **Use existing connection** pour réutiliser la connexion créée par le *tDropboxConnection*.

3. Dans le champ **Path**, saisissez le chemin d'accès vers le fichier dans lequel vous souhaitez écrire les données en le faisant précéder d'une barre oblique (/). Par exemple, saisissez `/architecture.png`.

4. Dans la zone **Upload mode**, sélectionnez **Rename if existing**.

5. Sélectionnez le bouton **Upload incoming content as file**. Cela fait apparaître le bouton **Edit schema** pour vous permettre de voir le schéma en lecture seule de ce composant.

**Exécuter le Job**

Appuyez sur **F6** pour exécuter ce Job.

Lorsque vous avez terminé, vérifiez les fichiers téléchargés dans le dossier spécifié de votre application Dropbox. Dans ce scénario, il s'agit du dossier *talenddrop*. 

---

![Image de Dropbox](image.png)
tDTDValidator

Ce composant permet de vérifier les données et la qualité de la structure du fichier traité.

Le composant tDTDValidator valide un fichier XML entrant en fonction d’une DTD et envoie les informations de validation à la sortie définie.

**Propriétés du tDTDValidator Standard**

Ces propriétés sont utilisées pour configurer le tDTDValidator s’exécutant dans le framework de Jobs Standard.

Le composant tDTDValidator Standard appartient à la famille XML.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th><strong>Schema et Edit Schema</strong></th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule. Il contient les informations standard de validation de fichier.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DTD file</strong></td>
<td>Chemin d’accès au fichier DTD de référence.</td>
</tr>
<tr>
<td><strong>XML file</strong></td>
<td>Chemin d’accès au fichier XML à valider.</td>
</tr>
</tbody>
</table>
| **If XML is valid, display** | **If XML is invalid, display**
  Saisissez un message à afficher dans la console **Run** basé sur le résultat de la comparaison. |
| **Print to console**       | Cochez cette case pour afficher le message de validation. |

**Advanced settings**

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

<table>
<thead>
<tr>
<th><strong>Global Variables</strong></th>
<th><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option. <strong>DIFFERENCE</strong> : résultat de la validation. Cette variable est une variable <strong>Flow</strong> et retourne une chaîne de caractères. <strong>VALID</strong> : résultat de la validation. Cette variable est une variable <strong>Flow</strong> et retourne un booléen.</th>
</tr>
</thead>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

#### Règle d’utilisation

Ce composant peut être utilisé en standalone mais il est généralement connecté à un composant de sortie pour rassembler les informations de validation.

### Scénario : Valider un fichier XML

Ce scénario décrit un Job validant le type de fichiers spécifié dans un dossier, affichant les résultats de validation dans la console de la vue **Run** et écrivant dans un fichier délimité les informations de log des fichiers invalides.

![Diagramme de flux pour la validation de fichiers XML](image)

**Valider un fichier XML**

**Procédure**

1. Cliquez et déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un **tFileList**, un **tDTDValidator**, un **tMap** et un **tFileOutputDelimited**.
2. Connectez le **tFileList** au **tDTDValidator** avec un lien **Iterate** et les composants restants avec des liens **Row Main**.
3. Paramétrez les propriétés du composant **tFileList** afin qu’il récupère les fichiers XML du dossier.
Cliquez sur le bouton [+] pour ajouter une ligne de masque de fichier et saisissez le masque suivant : ".*\.xml". Rappelez-vous que le code Java nécessite des guillemets doubles.

Paramétrez le chemin d'accès des fichiers XML à vérifier.

 Sélectionnez No dans la liste déroulante Case Sensitive.

4. Dans la vue Component du composant tDTDValidate, le schéma est en lecture seule puisqu'il contient des informations standard de log relatives au processus de validation.

Dans le champ Dtd file, parcourrez votre système jusqu'au fichier DTD à utiliser comme référence.

5. Dans le champ Xml file, appuyez sur Ctrl+Espace pour accéder à la liste des variables et sélectionnez la variable permettant de récupérer le chemin d'accès des fichiers : ((String)globalMap.get("tFileList_1_CURRENT_FILEPATH")).

6. Dans les divers messages à afficher dans la console Run, utilisez la variable jobName pour rappeler le nom du Job. Rappelez aussi le nom du fichier grâce à la variable : ((String)globalMap.get("tFileList_1_CURRENT_FILE")). Rappelez-vous que le code Java requiert des guillemets doubles.

 Cochez la case Print to Console.

7. Dans le composant tMap, glissez-déposez les informations du schéma standard que vous souhaitez passer au fichier de sortie.
8. Une fois le schéma de sortie défini, ajoutez une condition de filtre pour ne sélectionner que les informations de log lorsque le fichier XML est invalide.

Saisissez d'abord la valeur souhaitée pour cette variable, l'opérateur en fonction du type de données filtrées puis la variable devant correspondre à cette condition, ici : `0 == $row1[validate]`.

9. Puis (si cela n'est pas déjà effectué) connectez le **tMap** au **tFileOutputDelimited** via un lien **Row > Main**. Nommez-le **errorsOnly**.

10. Dans l'onglet **Basic settings** du composant **tFileOutputDelimited**, paramétrez le chemin d'accès de destination du fichier délimité, les séparateurs de champs et l'encodage.

11. Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

    Starting job tDTDValidator_oj at 14:28 14/10/2010
    [statistics] connecting to socket on port 3420
    [statistics] connected
    [Job tDTDValidator_oj] employee_1.xml is valid
    [Job tDTDValidator_oj] employee_2.xml is invalid
    [Job tDTDValidator_oj] employee_4.xml is invalid
    [Job tDTDValidator_oj] employee_5.xml is invalid
    [statistics] disconnected
    Job tDTDValidator_oj ended at 14:28 14/10/2010. [exit code=0]

Dans la console de la vue **Run**, les messages définis sont affichés pour chaque fichier invalide. Au même moment, le fichier de sortie est alimenté avec les données de log des fichiers invalides.
**tDynamoDBInput**

Ce composant récupère des données depuis une table Amazon DynamoDB et les envoie au composant suivant pour transformation.

**Propriétés du tDynamoDBInput Standard**

Ces propriétés sont utilisées pour configurer le tDynamoDBInput s’exécutant dans le framework de Jobs Standard.

Le composant tDynamoDBInput Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Access Key</th>
<th>Saisissez l’ID de la clé d’accès identifiant de manière unique un compte AWS. Pour plus d’informations concernant l’obtention de vos Access Key et Secret Key, consultez Obtention de vos clés d’accès AWS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secret Key</td>
<td>Saisissez la clé secrète d’accès (Secret Access Key), qui constitue, avec la clé d’accès, les informations d’authentification sécurisée. Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisées sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
</tbody>
</table>
| Assume role | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  
  - **Role ARN** : nom Amazon Resource Name (ARN) du rôle.  
  - **Role session name** : identifiant de la session du rôle.  
  - **Session duration (minutes)** : durée (en minutes) pour laquelle est active la session du rôle.  
  Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais). |
<table>
<thead>
<tr>
<th><strong>Use End Point</strong></th>
<th>Cochez cette case et dans le champ <strong>Server Url</strong> affiché, spécifiez l’URL du service Web du service de base de données DynamoDB.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Region</strong></td>
<td>Spécifiez la zone géographique AWS en sélectionnant le nom d’une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (&quot;us-east-1&quot; par exemple) dans la liste. Pour plus d’informations sur les zones géographiques AWS, consultez <a href="https://aws.amazon.com/fr/regions/">Régions et points de terminaison AWS</a>.</td>
</tr>
<tr>
<td><strong>Action</strong></td>
<td>Sélectionnez l’opération à effectuer dans la liste déroulante, soit <strong>Query</strong> soit <strong>Scan</strong>. Pour plus d’informations, consultez <em>Query and Scan Operations in DynamoDB</em> (en anglais).</td>
</tr>
<tr>
<td><strong>Schema et Edit schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <strong>No</strong> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre <a href="https://www.talend.com">Repository Content</a>.</td>
</tr>
<tr>
<td><strong>Table Name</strong></td>
<td>Spécifiez le nom des tables à interroger ou à scanner.</td>
</tr>
<tr>
<td><strong>Use advanced key condition expression</strong></td>
<td>Cochez cette case et dans le champ <strong>Advanced key condition expression</strong> affiché, spécifiez les expressions de condition clés utilisées pour déterminer les éléments à lire depuis la table ou l’index.</td>
</tr>
<tr>
<td><strong>Key condition expression</strong></td>
<td>Spécifiez les expressions de condition clés utilisées pour déterminer les éléments à lire. Cliquez sur le bouton [*] pour ajouter autant de lignes que nécessaire, chaque ligne correspondant à une expression de condition</td>
</tr>
</tbody>
</table>
clé, puis configurez les attributs suivants pour chaque expression :

- **Key Column** : saisissez le nom de la colonne clé.
- **Function** : sélectionnez la fonction de l’expression de condition clé.
- **Value1** : spécifiez la valeur utilisée dans l’expression de condition clé.
- **Value2** : spécifiez la seconde valeur utilisée dans l’expression de condition clé si nécessaire, selon la fonction que vous avez sélectionnée.

Notez que seuls les éléments qui remplissent toutes les conditions clés définies dans la table peuvent être retournés.

Cette table n’est pas disponible lorsque la case **Use advanced key condition expression** est cochée.

<table>
<thead>
<tr>
<th>Use filter expression</th>
<th>Cochez cette case pour utiliser l’expression de filtre pour l’opération de requête ou de scan.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use advanced filter expression</td>
<td>Cochez cette case et dans le champ <strong>Advanced filter expression</strong> affiché, spécifiez les expressions de filtre utilisées pour affiner les données après qu’elles soient interrogées ou scannées et avant qu’elles vous soient retournées. Cette case est disponible lorsque la case <strong>Use filter expression</strong> est cochée.</td>
</tr>
</tbody>
</table>
| Filter expression | Spécifiez les expressions de filtre utilisées pour affiner les résultats qui vous sont retournés. Cliquez sur le bouton [+] pour ajouter autant de lignes que nécessaire, chaque ligne correspondant à une expression de filtre, puis configurez les attributs suivants pour chaque expression :

  - **Column** : saisissez le nom de la colonne utilisée pour affiner les résultats.
  - **Function** : sélectionnez la fonction de l’expression de filtre.
  - **Value1** : spécifiez la valeur utilisée dans l’expression de filtre.
  - **Value2** : spécifiez la seconde valeur utilisée dans l’expression de filtre si nécessaire, selon la fonction que vous avez sélectionnée.

Notez que seuls les éléments qui remplissent toutes les conditions de filtre définies dans la table peuvent être retournés.

Cette table est disponible lorsque la case **Use filter expression** est cochée et que la case **Use advanced filter expression** est décochée.

| Value mapping | Spécifiez les valeurs factices pour les valeurs des attributs des expressions. |
### Name mapping

Spécifiez les paramètres substituables des noms d’attribut qui entrent en conflit avec les mots réservés DynamoDB.

- **name**: saisissez le nom de l’attribut qui entre en conflit avec un mot réservé DynamoDB.
- **placeholder**: spécifiez le paramètre substituable du nom d’attribut correspondant.

Pour plus d’informations, consultez [Expression de noms d’attributs](#).

### Advanced settings

#### STS Endpoint

Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.

Cette case est disponible uniquement lorsque la case **Assume role** est cochée.

#### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>QUERY</strong></td>
<td>requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le [Guide utilisateur du Studio Talend](#).
**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie. |

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tDynamoDBOutput**

Ce composant écrit des données dans une table Amazon DynamoDB.
Le tDynamoDBOutput crée, met à jour ou supprime des données dans une table Amazon DynamoDB.

**Propriétés du tDynamoDBOutput Standard**

Ces propriétés sont utilisées pour configurer le tDynamoDBOutput s’exécutant dans le framework de Jobs Standard.
Le composant tDynamoDBOutput Standard appartient à la famille Big Data.
Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Access Key</th>
<th>Saisissez l’ID de la clé d’accès identifiant de manière unique un compte AWS. Pour plus d’informations concernant l’obtention de vos Access Key et Secret Key, consultez Obtention de vos clés d’accès AWS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secret Key</td>
<td>Saisissez la clé secrète d’accès (Secret Access Key), qui constitue, avec la clé d’accès, les informations d’authentification sécurisée. Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
</tbody>
</table>
| Assume role | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  
- **Role ARN** : nom Amazon Resource Name (ARN) du rôle.  
- **Role session name** : identifiant de la session du rôle.  
- **Session duration (minutes)** : durée (en minutes) pour laquelle est active la session du rôle. |
<table>
<thead>
<tr>
<th>tDynamoDBOutput</th>
<th>Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use End Point</strong></td>
<td>Cochez cette case et dans le champ Server Url affiché, spécifiez l’URL du service Web du service de base de données DynamoDB.</td>
</tr>
<tr>
<td><strong>Region</strong></td>
<td>Spécifiez la zone géographique AWS en sélectionnant le nom d’une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (&quot;us-east-1&quot; par exemple) dans la liste. Pour plus d’informations sur les zones géographiques AWS, consultez Régions et points de terminaison AWS.</td>
</tr>
</tbody>
</table>
| **Action on table** | Sélectionnez une opération à effectuer sur la table définie, vous pouvez effectuer l’une des opérations suivantes :  
- **Default** : aucune opération n’est effectuée.  
- **Drop and create table** : la table est supprimée et créée à nouveau.  
- **Create table** : la table n’existe pas et est créée.  
- **Create table if does not exist** : la table est créée si elle n’existe pas.  
- **Drop table if exist and create** : la table est supprimée si elle existe déjà et créée à nouveau. |
| **Action on data** | Vous pouvez effectuer une des opérations suivantes sur les données de la table définie :  
- **Insert** : insérer de nouveaux éléments à partir du flux d’entrée.  
- **Update** : mettre à jour des éléments existants selon le flux d’entrée.  
- **Delete** : supprimer des éléments existants selon le flux d’entrée. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
| | Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma. |
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Table Name**
Spécifiez le nom de la table à écrire.

**Partition Key**
Spécifiez la clé de partition de la table spécifiée.

**Sort Key**
Spécifiez la clé de tri de la table spécifiée.

**Advanced settings**

**STS Endpoint**
Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.

Cette case est disponible uniquement lorsque la case **Assume role** est cochée.

**Read Capacity Unit**
Spécifiez le nombre d’unités de capacité de lecture. Pour plus d’informations, consultez Capacité de débit pour la lecture et l’écriture.

**Write Capacity Unit**
Spécifiez le nombre d’unités de capacité d’écriture. Pour plus d’informations, consultez Capacité de débit pour la lecture et l’écriture.

**tStatCatcher Statistics**
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Global Variables**

**Global Variables**

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

**QUERY** : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de fin dans un Job ou un sous-job et nécessite un lien d’entrée. |

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tEDIFACTtoXML

Ce composant transforme un fichier de message EDIFACT au format XML afin que les utilisateurs en aient une meilleure lisibilité et que la compatibilité avec les outils de traitement soit meilleure.

Le tEDIFACTtoXML lit un message EDIFACT des Nations Unies (Echange de Données Informatisées pour l'Administration, le Commerce et le Transport) et le transforme au format XML selon la version et la famille EDIFACT.

Propriétés du tEDIFACTtoXML Standard

Ces propriétés sont utilisées pour configurer le tEDIFACTtoXML s'exécutant dans le framework de Jobs Standard.

Le composant tEDIFACTtoXML Standard appartient à la famille XML.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est fixe et en lecture seule, avec une colonne : document.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDI filename</td>
<td>Chemin d'accès au fichier de message EDIFACT à transformer.</td>
</tr>
<tr>
<td>EDI version</td>
<td>Sélectionnez dans la liste la version d'EDIFACT du fichier d'entrée.</td>
</tr>
<tr>
<td>Ignore new line</td>
<td>Cochez cette case pour ignorer le retour chariot dans le fichier d'entrée.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est décochée par défaut et stoppe le Job en cas d'erreur. Cela permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Cochez cette case pour arrêter l'exécution du Job en cas d'erreur.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. |

Global Variables

| Global Variables | ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

---

**Utilisation**

| Règle d’utilisation | Ce composant doit être lié à un composant de sortie afin de rassembler les résultats de transformation. |

---

**Scénario : Passer d'EDIFACT à XML**

Ce scénario décrit un Job simple qui lit un fichier de message EDIFACT Customs Cargo (CUSCAR) et le sauvegarde en tant que fichier XML.

**Ajouter et relier les composants**

**Procédure**

1. Déposez un composant **tEDIFACTtoXML** et un **tFileOutputXML** de la Palette dans l’espace de modélisation graphique.
2. Connectez le composant **tEDIFACTtoXML** au **tFileOutputXML** à l’aide d’un lien **Row > Main**.

**Résultats**

![Diagramme de connexion des composants](image)

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le composant **tEDIFACTtoXML** afin d’afficher sa vue **Basic settings**.
2. Saisissez dans le champ **EDI filename** le chemin d’accès au fichier du message EDIFACT d’entrée ou parcourez votre système. 

Dans ce scénario, le fichier d’entrée est **99a_cuscar.edi**.

3. Dans la liste **EDI version**, sélectionnez la version d’EDIFACT du fichier d’entrée, **D99A** dans cet exemple.

4. Cochez la case **Ignore new line** afin d’ignorer les caractères de retour chariot dans le fichier d’entrée durant la transformation.

5. Laissez les autres paramètres tels qu’ils sont.

6. Double-cliquez sur le composant **tFileOutputXML** afin d’afficher sa vue **Basic settings**.

7. Dans le champ **File Name**, saisissez le chemin d’accès au fichier XML de sortie que vous souhaitez générer ou parcourez votre système.

Dans cet exemple, le fichier XML de sortie est **99a_cuscar.xml**.

8. Laissez les autres paramètres tels qu’ils sont.

**Enregistrer et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour enregistrer votre Job.
2. Appuyez sur **F6** pour exécuter le Job.

**Résultats**

Le fichier du message d’entrée EDIFACT CUSCAR est transformé au format XML et le fichier XML de sortie est généré comme défini.
**tELTGreenplumInput**

Ce composant ajoute autant de tables d’entrée (input) que nécessaire pour les requêtes Insert les plus complexes.

Les trois composants ELT Greenplum sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Greenplum, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Fournit le schéma de table à utiliser pour la requête SQL à exécuter.

**Propriétés du tELTGreenplumInput Standard**

Ces propriétés sont utilisées pour configurer le tELTGreenplumInput s’exécutant dans le framework de Jobs Standard.

Le composant tELTGreenplumInput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit schema | Un schéma est une description de lignes, il définit la nature le nombre de champs à traiter. Le schéma peut être Built-in ou distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
|                       | • View schema : sélectionnez cette option afin de voir le schéma.  
|                       | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
|                       | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].  
Default Table Name  |  Saisissez le nom de la table par défaut, entre guillemets doubles.

Default Schema Name  |  Saisissez le nom du schéma par défaut, entre guillemets doubles.

Advanced settings

| tStatCatcher Statistics  |  Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables


Utilisation

| Règle d’utilisation  |  L’utilisation du composant tELTGreenplumO utput est étroitement liée à celle des composants tELTGreenplumMap et tELTGreenplumInput. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.  

Remarque :  
Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

Scénarios associés

- Scénario : Mapper les données à l’aide d’une jointure implicite simple à la page 724
- Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- Scénario 2 : ELT utilisant une table Alias à la page 794
• Agréger des données Snowflake à l'aide de variables de contexte comme noms de tables et de connexion à la page 767
tELTGreenplumMap

Ce composant utilise les tables fournies en entrée pour alimenter les paramètres dans la commande mise en place. L'instruction peut inclure des joints internes ou externes qui peuvent être implémentés entre des tables ou entre une table et ses alias.

Les trois composants ELT Greenplum sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Greenplum, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTGreenplumMap permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.

Propriétés du tELTGreenplumMap Standard

Ces propriétés sont utilisées pour configurer le tELTGreenplumMap s'exécutant dans le framework de Jobs Standard.

Le composant tELTGreenplumMap Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</th>
</tr>
</thead>
</table>
|                           | Remarque :
|                           | Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :
|                           | 1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
|                           | 2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.
|                           | Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend. |

ELT Greenplum Map Editor

L’éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.
| Style link | Sélectionnez le type d’affichage des liens.  
**Auto** : par défaut, les liens entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de courbe.  
**Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne.  
**Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d’optimiser les performances. |
|---|---|
| Property type | Peut être **Built-in** ou **Repository**.  
**Built-in** : Propriétés utilisées ponctuellement.  
**Repository** : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |
| Host | Adresse IP du serveur de base de données. |
| Port | Numéro de port d’écoute du serveur de la base. |
| Database | Nom de la base de données |
| Username et Password | Données d’authentification de l’utilisateur de la base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| Advanced settings |  |
| Additional JDBC parameters | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n’est pas disponible lorsque vous utilisez l’option **Use an existing connection** dans les **Basic settings**. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau du composant. |
| Global Variables |  |
| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. |
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | L’utilisation du composant **tELTGreenplumMap** est étroitement liée à celle des composants **tELTGreenplumInput** et **tELTGreenplumOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*. |
Scénario : Mapper les données à l'aide d'une jointure implicite simple

Dans ce scénario, un composant tELTGreenplumMap est utilisé afin de récupérer les données de la table source employee_by_statecode, comparer sa colonne statecode à la table statecode et enfin mapper les colonnes désirées à partir des deux tables vers la table de sortie employee_by_state.

Avant l'exécution du Job, les trois tables employee_by_statecode, statecode et employee_by_state ressemblent à ceci :

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>statecode</th>
<th>state</th>
<th>statecode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Andy</td>
<td>86</td>
<td>China</td>
<td>86</td>
</tr>
<tr>
<td>3</td>
<td>Mike</td>
<td>39</td>
<td>Mexico</td>
<td>52</td>
</tr>
<tr>
<td>2</td>
<td>John</td>
<td>52</td>
<td>Italy</td>
<td>39</td>
</tr>
</tbody>
</table>

Déposer les composants

Procédure

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un tGreenplumConnection, deux tELTGreenplumInput, un tELTGreenplumMap, un tELTGreenplumOutput, un tGreenplumCommit, un tELTGreenplumInput et un tLogRow.

2. Renommez le composant tGreenplumConnection en connect_to_greenplum_host, les deux tELTGreenplumInput en employee+statecode et statecode, le tELTGreenplumMap en match+map, le tELTGreenplumOutput en map_data_output, le tGreenplumCommit en commit_to_host, le tGreenplumInput en read_map_output_table et le tLogRow en show_map_data.

3. Reliez le composant tGreenplumConnection au tELTGreenplumMap à l’aide d’un lien OnSubjobOk.
Reliez le composant tELTGreenplumMap au tGreenplumCommit à l’aide d’un lien OnSubjobOk.
Reliez le composant tGreenplumCommit au tGreenplumInput à l’aide d’un lien OnSubjobOk.

4. Reliez le composant tGreenplumInput au tLogRow à l’aide d’une connexion Row > Main.
Les deux tELTGreenplumInput et le composant tELTGreenplumOutput seront connectés plus tard au tELTGreenplumMap, une fois les tables correspondantes définies.
Configurer les composants

Procédure

1. Double-cliquez sur le composant tGreenplumConnection pour ouvrir sa vue Basic settings.

Dans les champs Host et Port, saisissez les variables de contexte pour le serveur Greenplum.
Dans le champ Database, saisissez la variable de contexte pour la base de données Greenplum.
Dans les champs Username et Password, saisissez les variables de contexte pour les paramètres d’authentification.
Pour plus d’informations sur les variables de contexte, consultez le Guide utilisateur du Studio Talend.


Dans le champ Default table name, saisissez le nom de la table source, nommée employee_by_statecode.
Cliquez sur le bouton [...] près du champ Edit schema pour ouvrir l’éditeur de schéma.

Cliquez sur le bouton [+] pour ajouter trois colonnes et appelez-les id, name et statecode, avec pour type respectivement INT4, VARCHAR, et INT4.
Cliquez sur **OK** pour fermer l’éditeur de schéma.
Reliez le **employee+statecode** au tELTGreenplumMap en utilisant le composant de sortie employee_by_statecode.

3. Double-cliquez sur le **statecode** pour ouvrir sa vue **Basic settings** dans l’onglet **Component**.

Dans le champ **Default table name**, saisissez le nom de la table, ici **statecode**.

4. Cliquez sur le bouton [...] près du champ **Edit schema** pour ouvrir l’éditeur de schéma.

Cliquez sur le bouton [+] pour ajouter deux colonnes et appelez-les **state** et **statecode**, avec pour type respectivement VARCHAR et INT4.

Cliquez sur **OK** pour fermer l’éditeur de schéma.

5. Cliquez le composant tELTGreenplumMap pour ouvrir sa vue **Basic settings** dans l’onglet **Component**.

Cochez la case **Use an existing connection**.

6. Cliquez sur le bouton [...] près du champ **ELT Greenplum Map Editor** pour ouvrir l’éditeur de mapping.
7. Cliquez sur le bouton [+] en haut à gauche pour ouvrir la fenêtre de sélection de table.

Sélectionnez les tables `employee_by_statecode` et `statecode` dans la liste et cliquez sur **Ok**.
Les tables apparaissent sur le panneau gauche de l’éditeur.

8. Dans le coin en haut à droite, cliquez sur le bouton [+] pour ajouter la table de sortie nommée `employee_by_state`.
Cliquez sur **OK** pour fermer l’éditeur de map.

9. Double-cliquez sur le **tELTGreenplumOutput** pour ouvrir sa vue **Basic settings** dans l’onglet **Component**.
Dans le champ **Default table name**, saisissez le nom de la table de sortie, ici *employee_by_state*.

10. Cliquez sur le bouton [...] près du champ **Edit schema** pour ouvrir l’éditeur de schéma.

Cliquez sur le bouton [+ ] pour ajouter trois colonnes et appelez-les respectivement *id*, *name* et *state*, avec pour type respectivement *INT4*, *VARCHAR*, et *VARCHAR*.

Cliquez sur **OK** pour fermer l’éditeur de schéma.

Reliez le composant **tELTGreenplumMap** au **tELTGreenplumOutput** en utilisant la table de sortie *employee_by_state*.

Cliquez sur **OK** dans la fenêtre pop-up pour récupérer le schéma du **tELTGreenplumOutput**.

La table de sortie *employee_by_state* partage maintenant le même schéma que celui du **tELTGreenplumOutput**.

11. Double-cliquez sur le composant **tELTGreenplumMap** pour ouvrir l’éditeur de map.

Déposez la colonne *statecode* de la table *employee_by_statecode* dans la colonne du même nom, dans la table *statecode*, afin de chercher les enregistrements des deux tables ayant les mêmes valeurs *statecode*. 
Déposez les colonnes *id* et *name* de la table *employee_by_statecode* ainsi que la colonne *statecode* de la table *statecode* dans les colonnes de même nom dans la table de sortie *employee_by_state*.

Cliquez sur OK pour fermer l’éditeur de mapping.

12. Double-cliquez sur le **tGreenplumInput** pour ouvrir sa vue **Basic settings** dans l’onglet **Component**.

![read_map_output_table(tGreenplumInput_1)](image)

Cochez la case **Use an existing connection**.

Dans le champ **Table name**, saisissez le nom de la table source, appelée *employee_by_state*.

Dans le champ **Query**, saisissez la commande "SELECT * FROM "employee_by_state\"".

13. Double-cliquez sur le composant **tLogRow** pour ouvrir sa vue **Basic settings** dans l’onglet **Component**.

![show_map_data(tLogRow_2)](image)

Dans la zone **Mode**, sélectionnez l’option **Table (print values in cells of a table)** pour une meilleure lisibilité.

**Exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** pour exécuter le Job.

```
<table>
<thead>
<tr>
<th>[show_map_data]</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

[statistics] disconnected
```
Comme affiché ci-dessus, les données sur les employés ont été écrites dans la table *employee_by_state*, qui présente des informations géographiques concernant les employés.

**Scénario associé**

- Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre à la page 790.
- Scénario 2 : ELT utilisant une table Alias à la page 794.
- Pour un scénario expliquant les sous-requêtes, consultez Scénario : Mapper des données à l’aide d’une sous-requête à la page 848.
- Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion à la page 767
tELTGreenplumOutput

Exécute les requêtes SQL Insert, Update et Delete dans la base de données Greenplum.

Les trois composants ELT Greenplum sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Greenplum, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tGreenplumOutput exécute les actions sur la table spécifiée et insère les données selon le schéma de sortie défini dans le ELT Mapper.

Propriétés du tELTGreenplumOutput Standard

Ces propriétés sont utilisées pour configurer le tELTGreenplumOutput s’exécutant dans le framework de Jobs Standard.

Le composant tELTGreenplumOutput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Action on data | Sur les données de la table définie, vous pouvez effectuer les opérations suivantes :
| | **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s’arrête.
| | **Update** : Met à jour des lignes existantes.
| | **Delete** : Supprime les entrées correspondantes au flux d’entrée. |

| Schema et Edit schema | Un schéma est une description de lignes, il définit la nature le nombre de champs à traiter. Le schéma peut être **Built-in** ou distant dans le **Repository**.
| | Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
| | • **View schema** : sélectionnez cette option afin de voir le schéma.
| | • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
| | • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
### Built-in
Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository

### Where clauses for (for UPDATE and DELETE only)
Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.

### Default Table Name
Saisissez le nom de la table par défaut, entre guillemets doubles.

### Default Schema Name
Saisissez le nom du schéma par défaut, entre guillemets doubles.

### Table name from connection name is variable
Cochez cette case lorsque le nom de la connexion à ce composant est configuré comme une variable, par exemple une variable de contexte.

### Use different table name
Cochez cette case afin de définir le nom d’une table de sortie différente, entre guillemets doubles, dans le champ Table name qui s’affiche.

### Mapping
 Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d’informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. |
| | NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier. |
| | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

## Utilisation

### Règle d'utilisation

L'utilisation du composant **tELTGreenplumOutput** est étroitement liée à celle des composants **tELTGreenplumMap** et **tELTGreenplumInput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

Remarque :

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

## Scénarios associés

- Scénario : Mapper les données à l'aide d'une jointure implicite simple à la page 724
- Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- Scénario 2 : ELT utilisant une table Alias à la page 794
- Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion à la page 767
tELTHiveInput

Ce composant réplique le schéma de la table d’entrée Hive, qui sera utilisé par le composant tELTHiveMap suivant.

Les trois composants ELT Hive sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Hive, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTHiveInput fournit, pour le composant tELTHiveMap qui suit, le schéma d’entrée de la table Hive à utiliser.

Propriétés du tELTHiveInput Standard

Ces propriétés sont utilisées pour configurer le tELTHiveInput s’exécutant dans le framework de Jobs Standard.

Le composant tELTHiveInput Standard appartient à la famille ELT.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
</tbody>
</table>
**Default table name**
Saisissez le nom du schéma, entre guillemets doubles.

**Default schema name**
Saisissez le nom du schéma de base de données auquel la table d’entrée utilisée est liée.

**Advanced settings**

**tStat Catcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

**Variables globales**

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

**Règle d’utilisation**

L’utilisation du composant *tELTHiveOutput* est étroitement liée à celle des composants *tELTHiveMap* et *tELTHiveInput*. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

Si le *Studio Talend* utilisé pour vous connecter à une base de données Hive fonctionne sous Windows, vous devez créer manuellement un dossier appelé *tmp* à la racine du disque où le *Studio Talend* est installé.

**Remarque :**
Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

**Utilisation avec Dataproc**

Les composants ELT Hive nécessitent l’installation de Tez dans le cluster Google Cloud Dataproc à utiliser.

- Utilisez l’action d’initialisation présentée dans la documentation Google Cloud Platform : *Apache Tez on Dataproc* (en anglais).
- Pour plus d’informations concernant le concept général des actions d’initialisation dans un cluster.
Scénarios associés

- Scénario : Effectuer une jointure sur les colonnes d'une table et les écrire dans Hive à la page 749

- Agréger des données Snowflake à l'aide de variables de contexte comme noms de tables et de connexion à la page 767

Google Cloud Dataproc, consultez la documentation Google associée : Initialization actions (en anglais).
tELTHiveMap

Ce composant construit graphiquement votre instruction SQL pour transformer les données.

Les trois composants ELT Hive sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Hive, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTHiveMap utilise les tables fournies en entrée pour alimenter les paramètres dans la commande mise en place. L’instruction peut inclure des joints internes ou externes qui peuvent être implémentés entre des tables ou entre une table et ses alias.

**Propriétés du tELTHiveMap Standard**

Ces propriétés sont utilisées pour configurer le tELTHiveMap s’exécutant dans le framework de Jobs Standard.

Le composant tELTHiveMap Standard appartient à la famille ELT.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

Configuration de la connexion :

- Lorsque vous utilisez ce composant avec Google Dataproc :

<p>| Cluster identifier | Saisissez l’ID de votre cluster Dataproc à utiliser. |
| Region | Saisissez les régions dans lesquelles sont utilisées les ressources de calcul et dans lesquelles sont stockées et traitées les données. Si vous n’avez pas besoin de spécifier une région en particulier, laissez la valeur par défaut global. Pour plus d’informations relatives aux régions disponibles et aux zones de chaque groupe de région, consultez Regions and Zones (en anglais). |
| Google Storage staging bucket | Comme un Job Talend nécessite ses fichiers .jar dépendants pour être exécuté, spécifiez le répertoire Google Storage dans lequel ces fichiers .jar sont transférés afin que votre Job accède à ces fichiers lors de l’exécution. Le répertoire à saisir doit se terminer par une barre oblique (/). Si le répertoire n’existe pas, un répertoire est créé à la volée mais le bucket à utiliser doit déjà exister. |</p>
<table>
<thead>
<tr>
<th><strong>Database</strong></th>
<th>Renseignez ce champ avec le nom de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Provide Google Credentials in file</strong></td>
<td>Lorsque vous lancez votre Job à partir d’une machine donnée sur laquelle Google Cloud SDK a été installé et vous a autorisé à utiliser vos identifiants de compte utilisateur pour accéder à Google Cloud Platform, ne cochez pas cette case. Dans cette situation, cette machine est souvent votre machine locale. Pour plus d’informations concernant le fichier Google Credentials, contactez l’administrateur de votre Google Cloud Platform ou consultez Google Cloud Platform Auth Guide (en anglais).</td>
</tr>
</tbody>
</table>

- Lorsque vous utilisez ce composant avec HDInsight :

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HDInsight configuration</strong></td>
<td>Saisissez les informations d’authentification du cluster HD Insight à utiliser.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Renseignez ce champ avec le nom de la base de données.</td>
</tr>
</tbody>
</table>

- Lorsque vous utilisez les autres distributions :

<table>
<thead>
<tr>
<th><strong>Connection mode</strong></th>
<th>Sélectionnez un mode de connexion dans la liste. Les options varient en fonction de la distribution que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hive server</strong></td>
<td>Sélectionnez le serveur Hive sur lequel vous souhaitez que le Job utilisant ce composant exécute des requêtes dans Hive.</td>
</tr>
</tbody>
</table>
La liste Hive server est disponible uniquement lorsque la distribution Hadoop à utiliser, par exemple HortonWorks Data Platform V1.2.0 (Bimota) supporte HiveServer2. Vous pouvez sélectionner HiveServer2 (Hive 2), le serveur supportant mieux les connexions simultanées de différents clients que HiveServer (Hive 1).


| Host | Adresse IP du serveur de la base de données. |
| Port | Numéro du port d’écoute du serveur de la base de données. |
| Database | Renseignez ce champ avec le nom de la base de données. |
| Username et Password | Données d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Use kerberos authentication | Si vous accédez au Metastore de Hive avec une sécurité Kerberos, cochez cette case et saisissez ensuite les paramètres appropriés dans les champs qui s’affichent.  
- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.  
  Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.  
Les valeurs des paramètres suivants peuvent être trouvées dans le fichier hive-site.xml du système Hive utilisé. |
1. **Hive principal** utilise la valeur de `hive.metastore.kerberos.principal`. C'est le principal du service du Metastore de Hive.

2. **HiveServer2 local user principal** utilise la valeur de `hive.server2.authentication.kerberos.principal`.

3. **HiveServer2 local user keytab** utilise la valeur de `hive.server2.authentication.kerberos.keytab`.

4. **Metastore URL** utilise la valeur de `javax.jdo.option.ConnectionURL`. C'est la chaîne JDBC de connexion au Metastore de Hive.

5. **Driver class** utilise la valeur de `javax.jdo.option.ConnectionDriverName`. C'est le nom du pilote de la connexion JDBC.

6. **Username** utilise la valeur de `javax.jdo.option.ConnectionUserName`. Ce paramètre, ainsi que le paramètre **Password**, sont utilisés pour les informations de connexion de l'utilisateur au Metastore de Hive.

7. **Password** utilise la valeur de `javax.jdo.option.ConnectionPassword`.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

---

### Use a keytab to authenticate

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l'aide d'un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est **user1** et le principal à utiliser est **guest**. Dans cette situation, assurez-vous que **user1** a les droits de lecture pour le fichier keytab à utiliser.

---

### Use SSL encryption

Cochez cette case pour activer la connexion chiffrée SSL ou TLS.

Les champs qui s'affichent ensuite fournissent les informations d'authentification :

- Dans le champ **Trust store path**, saisissez le chemin ou parcourez votre système jusqu'au fichier TrustStore à utiliser. Par défaut, les types TrustStore supportés sont **JKS** et **PKCS 12**.

- Pour saisir le mot de passe, cliquez sur le bouton `[...]` à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe.
entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

Cette fonctionnalité n’est disponible que pour HiveServer2 en mode **Standalone** pour les distributions suivantes :
- Hortonworks Data Platform 2.0 +
- Cloudera CDH4 +
- Pivotal HD 2.0 +
- Amazon EMR 4.0.0 +

### Set Resource Manager

Cochez cette case et, dans le champ qui s’affiche, saisissez l’emplacement du ResourceManager de votre distribution. Par exemple `tal-qall4.tale nd.lan:8050`.

Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d’un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du de l’exécution) :

1. Cochez la case **Set resourcemanager scheduler address** et saisissez l’adresse de l’ordonnanceur (Scheduler) dans le champ qui apparaît.

2. Cochez la case **Set jobhistory address** et saisissez l’emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory.

3. Cochez la case **Set staging directory** et saisissez le chemin d’accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l’exécution de programmes. Ce répertoire se trouve sous la propriété `yarn.app.mapreduce.am.staging-dir` dans les fichiers de configuration, notamment les fichiers `yarn-site.xml` et `mapred-site.xml` de votre distribution.

4. Allouez des volumes de mémoire aux calculs **Map** et **Reduce** et au service **ApplicationMaster** de YARN en cochant la case **Set memory** dans la vue **Advanced settings**.

5. Cochez la case **Set Hadoop user** et saisissez le nom de l’utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d’écriture, ce champ vous permet d’exécuter le Job directement avec l’utilisateur ayant les droits d’accès appropriés au fichier ou répertoire à traiter.

6. Cochez la case **Use datanode hostname** pour permettre au Job d’accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété `dfs.client.use.datanode.hostname` à `true`. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.
Pour plus d’informations concernant le framework Hadoop Map/Reduce, consultez le tutoriel Map/Reduce dans la documentation de Apache Hadoop : http://hadoop.apache.org (en anglais).

Set NameNode URI

Cochez cette case et, dans le champ qui s’affiche, saisissez l’URI du NameNode Hadoop, le nœud maître d’un système Hadoop. Par exemple, si vous avez choisi une machine nommée masternode en tant que NameNode, l’emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

Pour plus d’informations concernant le framework Hadoop Map/Reduce, consultez le tutoriel Map/Reduce dans la documentation de Apache Hadoop : http://hadoop.apache.org (en anglais).

Les autres propriétés :

| Property type | Peut être Built-In ou Repository.  
Built-In : propriétés utilisées ponctuellement.  
Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

ELT Hive Map editor

L’éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande Hive QL à exécuter. Le nom des colonnes
du schéma peut être différent du nom des colonnes dans la base de données.

Si vous utilisez des variables de contexte dans la colonne **Expression**, dans l’éditeur de mapping pour mapper les schémas d’entrée et de sortie, ajoutez des guillemets simples autour de ces variables de contexte, par exemple, "context.v_erpName".

### Style link

Sélectionnez le type d’affichage des liens.

- **Auto** : par défaut, les liens entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de courbe.
- **Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne.
- **Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d’optimiser les performances.

### Distribution

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option **Custom** vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre
connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend**. **Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

### Hive version

Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.

### Execution engine

Cochez cette case et, dans la liste déroulante, sélectionnez le framework à utiliser pour exécuter le Job.

Cette liste est disponible lorsque vous utilisez le mode **Embedded** pour la connexion et distribution Hive avec laquelle vous travaillez, parmi les suivantes :

- Hortonworks : V2.1 et V2.2.
- MapR : V4.0.1.
- **Custom** : cette option vous permet de vous connecter à une distribution supportant mais non officiellement supportée par **Talend**.

Avant d’utiliser Tez, vérifiez que votre cluster Hadoop supporte Tez. Vous devez configurer l’accès aux bibliothèques Tez correspondantes via la vue **Advanced settings** de ce composant.

Pour plus d’informations concernant Hive avec Tez, consultez la documentation Apache à l’adresse [https://cwiki.apache.org/confluence/display/Hive/Hive+on +Tez](https://cwiki.apache.org/confluence/display/Hive/Hive+on +Tez) (en anglais). Des exemples vous sont présentés.
afin d’expliquer comment Tez peut être utilisé pour optimiser les performances par rapport à MapReduce.

Lorsque vous souhaitez permettre aux composants Hive d’accéder à HBase :

Ces paramètres sont disponibles uniquement lorsque la case **Use an existing connection** est décochée.

| Store by HBase | Cochez cette case afin d’afficher les paramètres à configurer pour permettre aux composants Hive d’accéder aux tables HBase :
| | • Une fois l’accès configuré, vous pourrez utiliser, dans un **tHiveRow** et un **tHiveInput**, les instructions Hive QL permettant de lire et d’écrire des données dans HBase.
| | • Si vous utilisez l’authentification Kerberos, vous devez définir les principaux relatifs à HBase dans les champs correspondants qui sont affichés.
| | Pour plus d’informations à propos de cet accès concernant Hive et HBase, consultez la documentation de Apache Hive concernant l’intégration Hive/HBase.

| Zookeeper quorum | Saisissez le nom ou l’URL du service Zookeeper utilisé pour coordonner les transactions entre votre Studio et votre base de données. Notez que, lorsque vous configurez Zookeeper, vous pouvez avoir besoin de configurer explicitement la propriété **zookeeper.znode.parent** pour définir le chemin vers le nœud znode racine contenant tous les znodes créés et utilisés par votre base de données. Cochez la case la case **Set Zookeeper znode parent** afin de définir cette propriété.

| Zookeeper client port | Saisissez le numéro du port d’écoute client du service Zookeeper que vous utilisez.

| Define the jars to register for HBase | Cochez cette case pour afficher la table **Register jar for HBase**, dans laquelle vous pouvez enregistrer tout fichier Jar manquant, requis pour HBase, par exemple, Hive Storage Handler, enregistré par défaut avec votre installation Hive.

| Register jar for HBase | Cliquez sur le bouton [+] pour ajouter des lignes à la table, puis, dans la colonne **Jar name**, sélectionnez le(s) fichier(s) Jar à enregistrer. Dans la colonne **Jar path**, saisissez le chemin d’accès à ce(s) Jar(s).

### Advanced settings

| Tez lib | Choisissez comment accéder aux bibliothèques de Tez :
| | • **Auto install** : lors de l’exécution, le Job charge et déploie les bibliothèques de Tez fournies par le Studio dans le répertoire spécifié dans le champ **Install folder in HDFS**, par exemple, /tmp/usr/tez.
| | Si vous avez configuré la propriété **tez.lib.uris** dans la table des propriétés, ce répertoire écrase la valeur de la propriété lors de l’exécution. Les autres propriétés... |
- **Use exist** : le Job accède aux bibliothèques de Tez déjà déployées dans le cluster Hadoop à utiliser. Vous devez saisir le chemin d'accès pointant vers ces bibliothèques dans le champ **Lib path (folder or file)**.

- **Lib jar** : cette table s'affiche lorsque vous avez sélectionné **Auto install** dans la liste **Tez lib** et que vous utilisez une distribution personnalisée (**Custom**). Dans cette table, vous devez ajouter les bibliothèques de Tez à charger.

### Temporary path

Si vous ne souhaitez pas configurer le Jobtracker et le NameNode lorsque vous exécutez la requête `select * from your_table_name`, vous devez paramétrer un chemin d'accès temporaire. Par exemple, `/C:/select_all` sous Windows.

### Hadoop properties

Le **Studio Talend** utilise une configuration par défaut pour son moteur, afin d'effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l'exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

Pour plus d'informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d'Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d'exemple, les liens vers certaines propriétés sont listés ci-après :


### Hive properties

Le **Studio Talend** utilise la configuration par défaut pour son moteur afin d'effectuer des opérations dans un base de données Hive. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez cette table avec la (les)
propriété(s) à personnaliser. Ensuite, à l’exécution, la (les) propriété(s) personnalisée(s) écrasent celles par défaut. Pour plus d’informations concernant les propriétés dédiées à Hive, consultez https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration (en anglais).

- Si vous devez utiliser Tez pour exécuter votre Job Hive, ajoutez hive.execution.engine à la colonne Properties et Tez à la valeur Value, en entourant ces chaînes de caractères de guillemets doubles.
- Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste Property type, vous passiez de Repository à Built-in.

| Mapred job map memory mb et Mapred job reduce memory mb | Vous pouvez personnaliser les opérations map et reduce en cochant la case Set memory, pour configurer les allocations de mémoire pour ces opérations à effectuer par le système Hadoop. 
Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations map et reduce dans les champs Mapred job map memory mb et Mapred job reduce memory mb, respectivement. Par défaut, les valeurs sont toutes les deux 1000, ce qui est normalement adapté pour l’exécution de ces opérations. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Path separator in server</td>
<td>Laissez le champ Path separator in server tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d’autres termes, changez le séparateur si celui-ci n’est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau du composant.</td>
</tr>
<tr>
<td>Temporary path</td>
<td>Si vous ne souhaitez pas configurer le Jobtracker et le NameNode lorsque vous exécutez la requête select * from your_table_name, vous devez paramétrer un chemin d’accès temporaire. Par exemple, /C:/select_all sous Windows.</td>
</tr>
</tbody>
</table>

### Variables globales

| Variables globales | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |
Pour renseigner un champ ou une expression à l'aide
d’une variable, appuyez sur les touches **Ctrl+Espace**
pour accéder à la liste des variables. A partir de cette
liste, vous pouvez choisir la variable que vous souhaitez
utiliser.

Pour plus d’informations concernant les variables,
consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Le **tELTHiveMap** est généralement utilisé avec
d’autres composants, tels que le **tELTHiveInput** et le
**tELTHiveOutput**. Notez que le lien de sortie (output)
à utiliser avec ces deux composants doit respecter
strictement la syntaxe du nom de la table.

Si le Studio Talend utilisé pour vous connecter à une
base de données Hive fonctionne sous Windows, vous
devrez créer manuellement un dossier appelé `tmp` à la
racine du disque où le Studio Talend est installé.

**Remarque :**
Les composants ELT ne traitent pas le flux
de données lui-même mais uniquement les
informations du schéma.

| Utilisation avec Dataproc | Les composants ELT Hive nécessitent l’installation de
Tez dans le cluster Google Cloud Dataproc à utiliser.
- Utilisez l’action d’initialisation présentée dans la
documentation Google Cloud Platform : Apache Tez
  on Dataproc (en anglais).
- Pour plus d’informations concernant le concept
général des actions d’initialisation dans un cluster
Google Cloud Dataproc, consultez la documentation
Google associée : Initialization actions (en anglais).

| Dynamic settings | Cliquez sur le bouton `[+]` pour ajouter une ligne à la
table. Dans le champ **Code**, saisissez une variable de
contexte afin de sélectionner dynamiquement votre co
nnexion à la base de données parmi celles prévues dans
votre Job. Cette fonctionnalité est utile si vous devez
accéder à plusieurs tables de bases de données ayant
la même structure mais se trouvant dans différentes
bases de données, en particulier lorsque vous travaillez
dans un environnement dans lequel vous ne pouvez
pas changer les paramètres de votre Job, par exem
ple lorsque votre Job doit être déployé et exécuté
indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement
lorsque la case **Use an existing connection** est cochée
dans la vue **Basic settings**. Lorsqu’un paramètre
dynamique est configuré, la liste Component List de la
vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres
dynamiques, consultez Scénario : Lire des données
dans des bases de données à l’aide de connexions
Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Scénario : Effectuer une jointure sur les colonnes d’une table et les écrire dans Hive

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario utilise un Job comprenant quatre composants afin d’effectuer une jointure sur les colonnes sélectionnées de deux tables Hive et de les écrire dans une autre table Hive.
Préparer les tables Hive

Procédure

1. Créez la table Hive dans laquelle vous souhaitez écrire des données. Dans ce scénario, cette table est nommée `agg_result` et vous pouvez la créer à l'aide de l'instruction suivante dans le `tHiveRow` :

   ```
   create table agg_result (id int, name string, address string, sum1 string, postal string, state string, capital string, mostpopulouscity string) partitioned by (type string) row format delimited fields terminated by ';' location '/user/ychen/hive/table/agg_result'
   ```

   Dans cette instruction, `/user/ychen/hive/table/agg_result` est le répertoire utilisé pour stocker la table créée dans HDFS. Vous devez le remplacer par le répertoire à utiliser dans votre environnement.

   Pour plus d'informations concernant le `tHiveRow`, consultez `tHiveRow` à la page 1731.

2. Créez deux tables d’entrée Hive contenant les colonnes sur lesquelles vous souhaitez effectuer une jointure et que vous souhaitez agréger dans la table Hive de sortie, `agg_result`. Les instructions à utiliser sont :

   ```
   create table customer (id int, name string, address string, idState int, id2 int, regTime string, registerTime string, sum1 string, sum2 string) row format delimited fields terminated by ';' location '/user/ychen/hive/table/customer'
   ```

   ```
   create table state_city (id int, postal string, state string, capital int, mostpopulouscity string) row format delimited fields terminated by ';' location '/user/ychen/hive/table/state_city'
   ```

3. Utilisez le `tHiveRow` pour charger les données dans les deux tables d’entrée, `customer` et `state_city`. Les instructions à utiliser sont :

   ```
   "LOAD DATA LOCAL INPATH 'C:/tmp/customer.csv' OVERWRITE INTO TABLE customer" et "LOAD DATA LOCAL INPATH 'C:/tmp/State_City.csv' OVERWRITE INTO TABLE state_city"
   ```

Les deux fichiers, `customer.csv` et `State_City.csv`, sont deux fichiers locaux créés pour ce scénario. Vous devez créer vos fichiers afin de fournir les données aux tables Hive d’entrée. Le schéma des données de chaque fichier doit être identique à la table correspondante.

   Vous pouvez utiliser le `tRowGenerator` et le `tFileOutputDelimited` pour créer facilement les deux fichiers. Pour plus d’informations concernant ces composants, consultez `tRowGenerator` à la page 3478 et `tFileOutputDelimited` à la page 1169.

   Pour plus d’informations concernant le langage de requêtes Hive, consultez https://cwiki.apache.org/confluence/display/Hive/LanguageManual (en anglais).
Lier les composants

Procédure

1. Dans la perspective Integration du Studio Talend, créez un Job vide depuis le nœud Job Designs dans la vue Repository.
   Pour plus d’informations concernant la création d’un Job, consultez le Guide utilisateur du Studio Talend.
2. Déposez deux composants tELTHiveInput, un tELTHiveMap et un tELTHiveOutput dans l’espace de modélisation graphique.
3. Reliez les composants à l’aide de liens Row > Main.
   A chaque connexion de deux composants, un assistant s’ouvre et vous demande de nommez le lien que vous créez. Ce nom doit être le même que celui de la table Hive à traiter par le composant actif. Dans ce scénario, les tables d’entrée gérées par les deux composants tELTHiveInput sont customer et state_city et la table de sortie traitée par le tELTHiveOutput est agg_result.

Configurer les schémas d’entrée

Procédure

1. Double-cliquez sur le composant tELTHiveInput utilisant le lien customer afin d’ouvrir sa vue Component.
2. Cliquez sur le bouton [...] à côté du champ Edit schema pour ouvrir l’éditeur du schéma
3. Cliquez sur le bouton autant de fois que nécessaire pour créer des colonnes. Renommez-les de la même manière que dans le schéma de la table customer précédemment créée dans Hive.
4. Dans le champ **Default table name**, saisissez le nom de la table d’entrée, *customer*, à traiter par ce composant.

5. Double-cliquez sur le second *tELTHiveInput* utilisant le lien *state_city* pour ouvrir sa vue **Component**.

6. Cliquez sur le bouton [*...*] à côté du champ **Edit schema** pour ouvrir l’éditeur du schéma.

7. Cliquez sur le + autant de fois que nécessaire pour ajouter des colonnes. Renommez-les de la même manière que dans le schéma de la table *state_city* précédemment créée dans Hive.
8. Dans le champ Default table name, saisissez le nom de la table d’entrée, state_city, à traiter par ce composant.

mapper les schémas d’entrée et de sortie

Configurer la connexion à Hive

Procédure

1. Cliquez sur le composant tELTHiveMap, puis cliquez sur l’onglet Component pour ouvrir la vue correspondante.

2. Dans la zone Version, sélectionnez la distribution d’Hadoop que vous utilisez, ainsi que la version de Hive.

3. Dans la liste Connection mode, sélectionnez le mode de connexion à utiliser. Si votre distribution est HortonWorks, ce mode est Embedded uniquement.
4. Dans les champs Host et Port, saisissez les informations d’authentification pour que le composant se connecte à Hive. Dans cet exemple, l’hôte est talend-hdp-all et le port est 9083.

Mapper les schémas

Procédure

1. Cliquez sur le bouton ELT Hive Map Editor afin de mapper les schémas

2. Du côté de l’entrée (à gauche), cliquez sur le bouton Add alias pour ajouter la table à utiliser.

3. Dans la fenêtre qui s’ouvre, sélectionnez la table customer, puis cliquez sur OK.
4. Répétez l’opération pour sélectionner la table state_city.
5. Glissez-déposez la colonne idstate de la table customer dans la colonne id de la table state_city. Une jointure Inner Join est automatiquement créée.
6. Du côté de la sortie (à droite), la table agg_result est vide. Cliquez sur le bouton , en bas, pour ajouter autant de colonnes que nécessaire. Nommez-les de la même façon que dans le schéma de la table agg_result précédemment créée dans Hive.
Remarque :
La colonne type est la colonne partitionnée de la table *agg_result* et ne doit pas être dupliquée dans ce schéma. Pour plus d’informations concernant la partition de colonnes dans des tables Hive, consultez le manuel de Hive.

7. De la table *customer*, glissez-déposez les colonnes *id, name, address* et *sum1* dans les colonnes correspondantes de la table *agg_result*.

8. De la table *state_city*, glissez-déposez les colonnes *postal, state, capital* et *mostpopuloucity* dans les colonnes correspondantes de la table *agg_result*.

Dans ce scénario, les variables de contexte ne sont pas utilisées dans la colonne Expression dans l’éditeur de mapping. Si vous utilisez des variables de contexte entourez-les de guillemets simples. Par exemple :

- `'context.gv_rootPid'`
- `'root.execn_pid'`
- `'context.v_currentInd'`
- `'curr_ind'`
- `'context.v_erpName'`
- `'erp_name'`
- `'context.v_instance'`
- `'erp_instance'`

9. Cliquez sur OK pour valider ces modifications.

**Configurer le schéma de sortie**

**Procédure**

1. Double-cliquez sur le composant tELTHiveOutput pour ouvrir sa vue Component.
2. Si le schéma de ce composant est différent de celui du composant précédent, une icône d’avertissement apparaît. Dans ce cas, cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent. L’icône disparaît.

3. Dans le champ **Default table name**, saisissez le nom de la table de sortie dans laquelle vous souhaitez écrire les données. Dans ce scénario, la table est *agg_result*.

4. Sous la table **Field partition**, cliquez sur le bouton pour ajouter une ligne. Cela vous permet d’écrire des données dans la colonne de partition de la table *agg_result*.
   
   Cette colonne de partition a été définie lors de la création de la table *agg_result* à l’aide de partitioned by (type string), dans l’instruction `Create` précédemment présentée. Cette colonne de partition est *type*, elle décrit le type des clients.

5. Dans la colonne **Partition column**, saisissez *type* sans guillemet. Dans la colonne **Partition value**, saisissez *prospective* entre guillemets simples.

**Exécuter le Job**

**Procédure**

Appuyez sur **F6** pour exécuter le Job.

**Résultats**

Vérifiez la table *agg_result* dans Hive, en utilisant, par exemple,

```
select * from agg_result;
```
Cette capture d'écran ne montre qu'une partie de la table. Vous pouvez constater que les colonnes d'entrée sélectionnées sont agrégées et écrites dans la table `agg_result`. La colonne de partition est remplie avec la valeur `prospective`.

**Scénarios associés**

- **Scénario : Effectuer une jointure sur les colonnes d'une table et les écrire dans Hive** à la page 749

- **Scénario : Mapper des données à l'aide d'une sous-requête** à la page 848, scénario utilisant une sous-requête

- **Agréger des données Snowflake à l'aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTHiveOutput

Ce composant s'utilise avec le tELTHiveMap afin d'écrire des données dans la table Hive.

Les trois composants ELT Hive sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Hive, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTHiveOutput exécute la requête construite par le composant tELTHiveMap précédent afin d’écrire des données dans la table Hive spécifiée.

Propriétés du tELTHiveOutput Standard

Ces propriétés sont utilisées pour configurer le tELTHiveOutput s’exécutant dans le framework de Jobs Standard.

Le composant tELTHiveOutput Standard appartient à la famille ELT.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Action on data | Sélectionnez l’action à effectuer sur les données à écrire dans la table Hive.
|                | Avec l’option Insert, les données à écrire dans la table Hive sont écrites à la suite des données existantes, s’il y en a. |
| Schema         | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. |
| Built-In       | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| Repository     | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |
| Edit schema    | Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|                | • View schema : sélectionnez cette option afin de voir le schéma. |
|                | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales. |
|                | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez |
propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th>Default table name</th>
<th>Saisissez le nom de la table par défaut, entre guillemets doubles.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default schema name</td>
<td>Saisissez le nom du schéma de base de données par défaut auquel la table de sortie utilisée est liée.</td>
</tr>
<tr>
<td>Table name from connection name is variable</td>
<td>Cochez cette case lorsque le nom de la connexion à ce composant est configuré comme une variable, par exemple une variable de contexte.</td>
</tr>
<tr>
<td>Use different table name</td>
<td>Cochez cette case pour définir un nom de table différent. Saisissez-le, entre guillemets doubles, dans le champ <strong>Table name</strong> qui apparaît. Si cette table est liée à un schéma de base de données différent de celui par défaut, vous devez également saisir le nom de ce schéma de base de données. La syntaxe est la suivante : nom_du_schéma.nom_de_la_table.</td>
</tr>
<tr>
<td>The target table uses the Parquet format</td>
<td>Si la table dans laquelle vous devez écrire les données est une table Parquet, cochez cette case. Dans la liste Compression qui apparaît, sélectionnez le mode de compression à utiliser pour gérer le fichier Parquet. Le mode par défaut est <strong>Uncompressed</strong>.</td>
</tr>
<tr>
<td>Field Partition</td>
<td>Dans la colonne <strong>Partition Column</strong>, saisissez le nom, sans guillemet, de la colonne de partition de la table Hive dans laquelle vous souhaitez écrire les données. Dans la colonne <strong>Partition Value</strong>, saisissez la valeur que vous souhaitez utiliser, entre guillemets simples, pour la colonne de partition correspondante.</td>
</tr>
<tr>
<td>Mapping</td>
<td>Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d’informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Variables globales**

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. |
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | L’utilisation du composant tELTHiveOutput est étroitement liée à celle des composants tELTHiveMap et tELTHiveInput. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

Si le Studio Talend utilisé pour vous connecter à une base de données Hive fonctionne sous Windows, vous devez créer manuellement un dossier appelé *tmp* à la racine du disque où le Studio Talend est installé.

**Remarque :** Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.


• Utilisez l’action d’initialisation présentée dans la documentation Google Cloud Platform : Apache Tez on Dataproc (en anglais).

• Pour plus d’informations concernant le concept général des actions d’initialisation dans un cluster Google Cloud Dataproc, consultez la documentation Google associée : Initialization actions (en anglais). |

### Scénarios associés

- Scénario : Effectuer une jointure sur les colonnes d’une table et les écrire dans Hive à la page 749
- Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion à la page 767
**tELTJDBCInput**

Ce composant ajoute autant de tables d’entrée que nécessaire pour exécuter une instruction SQL.

Les trois composants ELT JDBC sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données JDBC, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Il est recommandé d’utiliser les composants ELT spécifiques à un type de base de données (s’il y en a), au lieu des composants ELT JDBC. Par exemple, pour Teradata, il est recommandé d’utiliser les composants **tELTTeradataInput**, **tELTTeradataMap** et **tELTTeradataOutput**.

**Propriétés du tELTJDBCInput Standard**

Ces propriétés sont utilisées pour configurer le tELTJDBCInput s’exécutant dans le framework de Jobs Standard.

Le composant tELTJDBCInput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

**Basic settings**

| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l’utiliser dans la commande SQL Insert.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

| **Built-in** | Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |

**Default Table Name**
Saisissez le nom de la table par défaut.

**Default Schema Name**
Saisissez le nom du schéma par défaut.

**Mapping**
Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d’informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

**Global Variables**

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| Règle d’utilisation | L’utilisation du composant tELTJDBCInput est étroitement liée à celle du tELTJDBCMap. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. **Remarque :** Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma. |

**Scénarios associés**

- **Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre** à la page 790
• Scénario 2 : ELT utilisant une table Alias à la page 794
• Agréger des données Snowflake à l'aide de variables de contexte comme noms de tables et de connexion à la page 767
tELTJDBCMap

Ce composant utilise les tables fournies en entrée pour alimenter les paramètres nécessaires à l’exécution de la commande SQL construite. La commande peut contenir des jointures internes comme externes entre différentes tables ou entre une table et ses alias.

Les trois composants ELT JDBC sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données JDBC, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Il est recommandé d’utiliser les composants ELT spécifiques à un type de base de données (s’il y en a), au lieu des composants ELT JDBC. Par exemple, pour Teradata, il est recommandé d’utiliser les composants tELTTeradataInput, tELTTeradataMap et tELTTeradataOutput.

Propriétés du tELTJDBCMap Standard

Ces propriétés sont utilisées pour configurer le tELTJDBCMap s’exécutant dans le framework de Jobs Standard.

Le composant tELTJDBCMap Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Remarque :</td>
</tr>
<tr>
<td></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
</tr>
<tr>
<td></td>
<td>2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

<p>| ELT JDBC Map Editor | L’éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données. |</p>
<table>
<thead>
<tr>
<th>Style link</th>
<th>Sélectionnez le type d’affichage des liens.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>par défaut, les liens entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de courbe.</td>
</tr>
<tr>
<td>Bezier curve</td>
<td>les liens entre les schémas et les paramètres du service Web sont en forme de ligne.</td>
</tr>
<tr>
<td>Line</td>
<td>les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d’optimiser les performances.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
</tbody>
</table>


| Driver JAR         | Renseignez cette table afin de charger les Jar des pilotes requis. Pour ce faire, cliquez sur le bouton [+] sous la table, pour ajouter autant de lignes que nécessaire, chaque ligne pour un Jar de pilote. Sélectionnez la cellule et cliquez sur le bouton [...] à droite de la cellule pour ouvrir l’assistant Select Module, dans lequel vous pouvez sélectionner le Jar souhaité. Par exemple, le Jar du pilote RedshiftJDBC41-1.1.13.1013.jar pour la base de données Redshift. |

| Class name         | Saisissez entre guillemets doubles le nom de la classe pour le pilote spécifié. Par exemple, pour le pilote RedshiftJDBC41-1.1.13.1013.jar, le nom à saisir est com.amazon.redshift.jdbc41.Driver. |

| Username et Password | Données d’authentification de l’utilisateur de la base. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |

| Mapping             | Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d’informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types. |

| Advanced settings   | Additional JDBC parameters | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option |
| Règle d’utilisation | L’utilisation du composant `tELTJDBCMap` est étroitement liée à celle des composants `tELTJDBCInput` et `tELTJDBCOutput`. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.  

**Remarque** :
Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable. |
Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion

Ce scénario présente un exemple d’agrégation de données Snowflake à partir de deux tables source STUDENT et TEACHER vers une table cible FULLINFO, à l’aide des composants JDBC ELT. Dans cet exemple, tous les noms des tables de sortie et des connexions sont configurés comme variables de contexte.

Créer un Job pour agrégérer des données Snowflake

Avant de commencer

- Un nouveau Job doit avoir été créé et les variables de contexte SourceTableS avec la valeur STUDENT, SourceTableT avec la valeur TEACHER et TargetTable avec la valeur FULLINFO doivent avoir été ajoutées au Job. Pour plus d’informations concernant l’utilisation des variables de contexte, consultez la documentation relatives aux contextes et aux variables.
• La table source **STUDENT** avec trois colonnes, **SID** et **TID** de type NUMBER(38,0) et **SNAME** de type VARCHAR(50) doit avoir été créée dans Snowflake et les données suivantes doivent avoir été écrites dans la table.

```
#SID;SNAME;TID
11;Alex;22
12;Mark;23
13;Stephane;21
14;Cedric;22
15;Bill;21
16;Jack;23
17;John;22
18;Andrew;23
```

• La table source **TEACHER** avec trois colonnes, **TID** de type NUMBER(38,0), **TNAME** et **TPHONE** de type VARCHAR(50), doit avoir été créée dans Snowflake et les données suivantes doivent avoir été écrites dans la table.

```
#TID;TNAME;TPHONE
21;Peter;+86 15812343456
22;Michael;+86 13178964532
23;Candice;+86 13923187456
```

**Procédure**

1. Ajoutez un composant **tSnowflakeConnection**, un **tSnowflakeClose**, deux **tELTJDBCInput**, un **tELTJDBCMap** et un **tELTJDBCOutput** à votre Job.

2. Dans la vue **Basic setting** du premier **tELTJDBCInput**, saisissez le nom de la première table source dans le champ **Default Table Name**. Dans cet exemple, saisissez la variable de contexte `context.SourceTableS`.

3. Répétez l’opération afin de configurer la valeur du nom de la table par défaut pour le second **tELTJDBCInput** et pour le **tELTJDBCOutput** à `context.SourceTableT` et `context.TargetTable` respectivement.

4. Reliez le premier **tELTJDBCInput** au **tELTJDBCMap** à l’aide d’un lien **Link > context.SourceTableS (Table)**.

5. Reliez le second **tELTJDBCInput** au **tELTJDBCMap** à l’aide d’un lien **Link > context.SourceTableT (Table)**.

6. Reliez le **tELTJDBCMap** au **tELTJDBCOutput** à l’aide d’un lien **Link > "New Output" (Table)**. Le lien sera automatiquement renommé en `context.TargetTable (Table)`.

7. Reliez le **tSnowflakeConnection** au **tELTJDBCMap** à l’aide d’un lien **Trigger > On Subjob Ok**.

8. Répétez l’opération pour relier le **tELTJDBCMap** au **tSnowflakeClose**.
Se connecter à Snowflake

Configurez le composant `tSnowflakeConnection` pour vous connecter à Snowflake.

**Procédure**

1. Double-cliquez sur le `tSnowflakeConnection` pour ouvrir sa vue `Basic settings`.
2. Dans le champ `Account`, saisissez le nom du compte assigné par Snowflake.
3. Dans les champs `User Id` et `Password`, saisissez les informations d’authentification.
   Notez que cet ID utilisateur est votre identifiant de connexion. Si vous ne connaissez pas votre identifiant, contactez l’administrateur de votre système Snowflake.
4. Dans le champ `Warehouse`, saisissez le nom de l’entrepôt de données à utiliser dans Snowflake.
5. Dans le champ `Schema`, saisissez le nom du schéma de la base de données à utiliser.
6. Dans le champ `Database`, saisissez le nom de la base de données à utiliser.

Configurer les composants d’entrée pour agréger les données Snowflake

**Procédure**

1. Double-cliquez sur le premier composant `tELTJDBCInput` pour ouvrir sa vue `Basic settings`.
2. Cliquez sur le bouton `[…]` à côté du champ `Edit schema` et, dans la fenêtre qui s’ouvre, définissez le schéma en ajoutant trois colonnes, `SID` et `TID` de type `INT` et `SNAME` de type `VARCHAR`.
4. Répétez les étapes précédentes pour configurer le second `tELTJDBCInput` et définissez son schéma en ajoutant trois colonnes, `TID` de type `INT`, puis `TNAME` et `TPHONE` de type `VARCHAR`.

Configurer le composant de sortie pour agréger les données Snowflake

**Procédure**

1. Double-cliquez sur le composant `tELTJDBCOutput` pour ouvrir sa vue `Basic settings`.
2. Sélectionnez `Create table` dans la liste `Action on table` afin de créer la table cible.
3. Cochez la case `the Table name from connection name is variable`.

Configurer le composant de mapping pour agréger les données Snowflake

**Procédure**

1. Cliquez sur le composant `tELTJDBCMap` pour ouvrir sa vue `Basic settings`.
2. Cochez la case **Use an existing connection** et, dans la liste **Component List** qui s’affiche, sélectionnez le composant de connexion configuré pour ouvrir la connexion à Snowflake.

3. Sélectionnez **Mapping Snowflake** dans la liste **Mapping**.

4. Cliquez sur le bouton [...] à côté du champ **ELT JDBC Map Editor** pour ouvrir l’éditeur de mapping.

5. Ajoutez la première table d’entrée `context.SourceTableS` en cliquant sur le bouton [+] dans le coin supérieur gauche de l’éditeur et en sélectionnant le nom de la table dans la liste, dans la boîte de dialogue qui s’affiche.

6. Répétez l’opération pour ajouter la seconde table d’entrée `context.SourceTableT`.

7. Déposez la colonne `TID` de la première table d’entrée `context.SourceTableS` et déposez-la dans la colonne `TID` correspondante, dans la seconde table d’entrée `context.SourceTableT`.

8. Déposez toutes les colonnes de la table d’entrée `context.SourceTableS` dans la table de sortie `context.TargetTable`, dans le panneau supérieur droit.

9. Répétez l’opération pour déposer deux colonnes `TNAME` et `TPHONE` de la table d’entrée au bas de la table de sortie `context.SourceTableT`. Cela fait, cliquez sur **OK** pour fermer l’éditeur de mapping.

10. Cliquez sur le bouton **Sync columns** dans la vue **Basic settings** du composant **tELTJDBCOutput** afin de définir son schéma.

**Fermer la connexion à Snowflake**

Configurez le composant **tSnowflakeClose** pour fermer la connexion à Snowflake.

**Procédure**

1. Double-cliquez sur le composant **tSnowflakeClose** pour ouvrir l’onglet **Component**.

2. Sélectionnez le composant qui ouvre la connexion que vous avez besoin de fermer dans la liste déroulante **Connection Component**, `tSnowflakeConnection_1` dans cet exemple.

**Exécuter le Job pour agréger les données Snowflake**

**Procédure**

1. Appuyez sur Ctrl + S pour sauvegarder votre Job.

2. Appuyez sur F6 pour exécuter le Job.
Comme affiché ci-dessus, le Job est exécuté avec succès et huit lignes sont insérées dans la table cible.

Vous pouvez créer et exécuter un autre Job afin de récupérer les données de la table cible en utilisant les composants `tSnowflakeInput` et `tLogRow`. Vous pouvez constater que les données agrégées sont affichées dans la console, comme dans la capture d’écran ci-dessous.

<table>
<thead>
<tr>
<th>SID</th>
<th>SNAME</th>
<th>TID</th>
<th>TNAME</th>
<th>TPHONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Alex</td>
<td>22</td>
<td>Michael</td>
<td>+86 131789964532</td>
</tr>
<tr>
<td>12</td>
<td>Mark</td>
<td>23</td>
<td>Candle</td>
<td>+86 13923187456</td>
</tr>
<tr>
<td>13</td>
<td>Stephane</td>
<td>21</td>
<td>Peter</td>
<td>+86 15812343456</td>
</tr>
<tr>
<td>14</td>
<td>Cedric</td>
<td>22</td>
<td>Michael</td>
<td>+86 13178964532</td>
</tr>
<tr>
<td>15</td>
<td>Bill</td>
<td>21</td>
<td>Peter</td>
<td>+86 15912343456</td>
</tr>
<tr>
<td>16</td>
<td>Jack</td>
<td>23</td>
<td>Candle</td>
<td>+86 13923187456</td>
</tr>
<tr>
<td>17</td>
<td>John</td>
<td>22</td>
<td>Michael</td>
<td>+86 131789964532</td>
</tr>
<tr>
<td>18</td>
<td>Andrew</td>
<td>23</td>
<td>Candle</td>
<td>+86 13923187456</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant la récupération de données de Snowflake, consultez l’exemple Écrire et lire des données dans une table Snowflake à la page 3763.

**Scénarios associés**

- **Scénario 1** : Agréger les colonnes d’une table et appliquer un filtre à la page 790.
- **Scénario 2** : ELT utilisant une table Alias à la page 794.
- **Scénario** : Mapper des données à l’aide d’une sous-requête à la page 848, scénario utilisant une sous-requête
tELTJDBCOutput

Ce composant effectue l’action sélectionnée sur la table et insère les données selon le schéma de sortie défini dans l’ELT Mapper.

Les trois composants ELT JDBC sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données JDBC, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Il est recommandé d’utiliser les composants ELT spécifiques à un type de base de données (s’il y en a), au lieu des composants ELT JDBC. Par exemple, pour Teradata, il est recommandé d’utiliser les composants tELTTeradataInput, tELTTeradataMap et tELTTeradataOutput.

Propriétés du tELTJDBCOutput Standard

Ces propriétés sont utilisées pour configurer le tELTJDBCOutput s’exécutant dans le framework de Jobs Standard.

Le composant tELTJDBCOutput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Action on table | Sélectionnez une opération à effectuer sur la table définie, vous pouvez effectuer l’une des opérations suivantes :
|                | • None : aucune opération n’est effectuée.
|                | • Drop and create table : la table est supprimée et créée à nouveau.
|                | • Create table : la table n’existe pas et est créée.
|                | • Create table if does not exist : la table est créée si elle n’existe pas.
|                | • Drop table if exist and create : la table est supprimée si elle existe déjà et créée à nouveau.
|                | • Clear table : le contenu de la table est supprimé. Vous pouvez annuler cette opération.
|                | • Truncate table : le contenu de la table est supprimé. Vous ne pouvez pas annuler cette opération.

| Action on data | Sur les données de la table définie, vous pouvez effectuer les opérations suivantes :
|               | • Insert : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s’arrête.
|               | • Update : Met à jour des lignes existantes.
|               | • Delete : Supprime les entrées correspondantes au flux d’entrée.
### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

|---|---|

### Where clauses for (for UPDATE and DELETE only)

Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.

### Default Table Name

Saisissez le nom de la table par défaut, entre guillemets doubles.

### Default Schema Name

Saisissez le nom du schéma par défaut, entre guillemets doubles.

### Table name from connection name is variable

Cochez cette case lorsque le nom de la connexion à ce composant est configuré comme une variable, par exemple une variable de contexte.

### Use different table name

Cochez cette case afin de définir le nom d’une table de sortie différente, entre guillemets doubles, dans le champ **Table name** qui s’affiche.

### Mapping

Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d’informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types.
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau du composant. |

### Global Variables

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.  
**NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable *After* et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | L’utilisation du composant tELTJDBCOutput est étroitement liée à celle des composants tELTJDBCInput et tELTJDBCMap. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.  
Évitez d’utiliser les mots-clés de la base de données ou d’utiliser des caractères spéciaux dans les noms de tables/colonnes. Si vous le souhaitez, vous pouvez entourer le nom des tables/colonnes de guillemets doubles ",", pour voir si cela fonctionne. Par exemple, lorsque vous utilisez le mot-clé *number* comme nom de colonne dans une base de données Oracle, il est possible que la valeur de *Db Column* dans l’éditeur de schéma soit configurée à "*number". Cependant, cette solution ne fonctionne pas toujours. |

### Scénarios associés

- Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre à la page 790
• Scénario 2 : ELT utilisant une table Alias à la page 794

• Agréger des données Snowflake à l'aide de variables de contexte comme noms de tables et de connexion à la page 767
**tELTMSSqlInput**

Ce composant ajoute autant de tables que nécessaires dans une instruction Insert qui peut être complexe.

Les trois composants ELT MSSql sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données MS SQL, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTMSSqlInput fournit le schéma de la table d’entrée à la commande SQL qui sera exécutée.

**Propriétés du tELTMSSqlInput Standard**

Ces propriétés sont utilisées pour configurer le tELTMSSqlInput s’exécutant dans le framework de Jobs Standard.

Le composant tELTMSSqlInput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

**Basic settings**

| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l'utiliser dans la commande SQL Insert.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

| **Built-in** | Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

**Default Table Name**
Saisissez le nom de la table par défaut.

**Default Schema Name**
Saisissez le nom du schéma par défaut.

**Advanced settings**

- **tStatCatcher Statistics**
  Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.
  
  Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.
  
  Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

- **Règle d’utilisation**

  L’utilisation du composant tELTMSSqlInput est étroitement liée à celle du tELTMSSqlMap. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

  **Remarque :**
  Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

**Scénarios associés**

- **Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre** à la page 790
- **Scénario 2 : ELT utilisant une table Alias** à la page 794
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTMSSqlMap

Utilise les tables fournies en entrée pour alimenter les paramètres nécessaires à l’exécution de la commande SQL souhaitée. La commande peut contenir des jointures internes comme externes entre différentes tables ou entre une table et ses alias.

Les trois composants ELT MSSql sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données MS SQL, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTMSSqlMap permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.

Propriétés du tELTMSSqlMap Standard

Ces propriétés sont utilisées pour configurer le tELTMSSqlMap s’exécutant dans le framework de Jobs Standard.

Le composant tELTMSSqlMap Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
</tr>
<tr>
<td></td>
<td>2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

| ELT MSSql Map Editor | L’éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données. |
Style link

Sélectionnez le type d’affichage des liens.

**Auto** : par défaut, les liens entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de courbe.

**Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne.

**Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d’optimiser les performances.

Property type

Peut être **Built-in** ou **Repository**.

**Built-in** : Propriétés utilisées ponctuellement.

**Repository** : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.

Host

Adresse IP du serveur de base de données.

Port

Numéro de port d’écoute du serveur de la base.

Database

Nom de la base de données.

Username et Password

Données d’authentification de l’utilisateur de la base.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Advanced settings

**Additional JDBC parameters**

Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n’est pas disponible lorsque vous utilisez l’option **Use an existing connection** dans les **Basic settings**.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau du composant.

Global Variables

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace**.
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d'utilisation | L'utilisation du composant tELTMSSqlMap est étroitement liée à celle des composants tELTMSSqlInput et tELTMSSqlOutput. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**
Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutile.


**Scénarios associés**

- Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- Scénario 2 : ELT utilisant une table Alias à la page 794
- Scénario : Mapper des données à l’aide d’une sous-requête à la page 848, scénario utilisant une sous-requête
• Agréger des données Snowflake à l'aide de variables de contexte comme noms de tables et de connexion à la page 767
tELTMSSqlOutput

Exécute les instructions Insert, Delete et Update dans la base de données MSSql.

Les trois composants ELT MSSql sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données MS SQL, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTMSSqlOutput effectue l'action sur la table telle que spécifiée et insère les données selon le schéma de sortie défini dans le Mapper ELT.

Propriétés du tELTMSSqlOutput Standard

Ces propriétés sont utilisées pour configurer le tELTMSSqlOutput s'exécutant dans le framework de Jobs Standard.

Le composant tELTMSSqlOutput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Action on data | Sur les données de la table définie, vous pouvez effectuer les opérations suivantes :
|                | **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s'arrête.
|                | **Update** : Met à jour des lignes existantes.
|                | **Delete** : Supprime les entrées correspondantes au flux d’entrée. |

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**.
|                       | Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
|                       | • **View schema** : sélectionnez cette option afin de voir le schéma.
|                       | • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
|                       | • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. |
**Built-in**: Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

**Repository**: Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.

**Where clauses for (for UPDATE and DELETE only)**

Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.

**Default Table Name**

Saisissez le nom de la table par défaut, entre guillemets doubles.

**Default Schema Name**

Saisissez le nom du schéma par défaut, entre guillemets doubles.

**Table name from connection name is variable**

Cochez cette case lorsque le nom de la connexion à ce composant est configuré comme une variable, par exemple une variable de contexte.

**Use different table name**

Cochez cette case afin de définir le nom d’une table de sortie différente, entre guillemets doubles, dans le champ *Table name* qui s’affiche.

**Mapping**

Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d’informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.

**NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable *After* et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

| Règle d’utilisation | L’utilisation du composant **tELTMSSqlOutput** est étroitement liée à celle des composants **tELTMSSqlInput** et **tELTMSSqlMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**
Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

| Limitation | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center ([https://help.talend.com](https://help.talend.com)).

**Scénarios associés**

- **Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre** à la page 790
- **Scénario 2 : ELT utilisant une table Alias** à la page 794
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
**tELTMysqlInput**

Ce composant ajoute autant de tables que nécessaires dans une commande Insert qui peut être complexe.

Les trois composants ELT MySQL sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données MySQL, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTMysqlInput fournit le schéma de la table d’entrée à la commande SQL qui sera exécutée.

**Propriétés du tELTMysqlInput Standard**

Ces propriétés sont utilisées pour configurer le tELTMysqlInput s’exécutant dans le framework de Jobs Standard.

Le composant tELTMysqlInput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

**Basic settings**

| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l’utiliser dans la commande SQL Insert. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
| | • **View schema** : sélectionnez cette option afin de voir le schéma.
| | • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
| | • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
| **Built-in** | Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend .
**Default Table Name**

Saisissez le nom de la table par défaut, entre guillemets doubles.

---

### Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau du composant.

---

### Global Variables

**Global Variables**

| ERROR_MESSAGE | message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

---

### Utilisation

**Règle d’utilisation**

L’utilisation du composant tELTMysqlInput est étroitement liée à celle du tELTMysqlMap. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

---

### Scénarios associés

- **Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre** à la page 790
- **Scénario 2 : ELT utilisant une table Alias** à la page 794
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTMysqlMap

Ce composant utilise les tables fournies en entrée pour alimenter les paramètres nécessaires à l'exécution de la commande SQL souhaitée. La commande peut contenir des jointures internes comme externes entre différentes tables ou entre une table et ses alias.

Les trois composants ELT MySQL sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données MySQL, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTMysqlMap permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.

Propriétés du tELTMysqlMap Standard

Ces propriétés sont utilisées pour configurer le tELTMysqlMap s'exécutant dans le framework de Jobs Standard.

Le composant tELTMysqlMap Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

<table>
<thead>
<tr>
<th>Basic settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DB Version</strong></td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. **Au niveau parent,** enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. **Au niveau enfant,** utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

| **ELT Mysql Map editor** | L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du |
schéma peut être différent du nom des colonnes dans la base de données.

<table>
<thead>
<tr>
<th>Style link</th>
<th>Sélectionnez le type d’affichage des liens.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Auto</strong> : par défaut, les liens entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de courbe.</td>
<td></td>
</tr>
<tr>
<td><strong>Bezier curve</strong> : les liens entre les schémas et les paramètres du service Web sont en forme de ligne.</td>
<td></td>
</tr>
<tr>
<td><strong>Line</strong> : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d’optimiser les performances.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
</tbody>
</table>

| Host | Adresse IP du serveur de base de données. |
| Port | Numéro de port d’écoute du serveur de la base. |
| Database | Nom de la base de données. |

<table>
<thead>
<tr>
<th>Username et Password</th>
<th>Données d’authentification de l’utilisateur de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
<td></td>
</tr>
</tbody>
</table>

| Preview | L’aperçu est un cliché des données traitées dans le Mapper. Il apparaît lorsque les propriétés du Mapper ont été complétées. La synchronisation de l’aperçu n’est disponible qu’après avoir enregistrer les modifications. |

**Advanced settings**

| tStatCatcher Statistics | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

**Global Variables**

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. |
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

**Utilisation**

### Règle d’utilisation

L’utilisation du composant **tELTMysqlMap** est étroitement liée à celle des composants **tELTMysqlInput** et **tELTMysqlOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

### Dynamic settings

Cliquez sur le bouton `[+]` pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez  **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et  **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.
Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre

Le scénario suivant décrit un Job rassemblant plusieurs schémas d'une table de base de données en entrée puis utilise une clause dans la requête SQL afin de filtrer les données en sortie.

- Cliquez et déposez les composants suivants de la Palette dans l'espace de modélisation graphique : trois tELTMysqlInput, un tELTMysqlMap, et un tELTMysqlOutput.
- Double-cliquez sur le premier tELTMysqlInput afin d'afficher sa vue Basic settings.

Dans la liste Schema, sélectionnez Repository, cliquez sur le bouton [...] à côté du champ Edit schema. Dans la boîte de dialogue [Repository Content], sélectionnez votre connexion à la base de données ainsi que le schéma souhaité.

Le nom du schéma sélectionné apparaît automatiquement dans le champ Default Table Name. Dans ce scénario, la connexion à la base de données est Talend_MySQL et le schéma pour le premier composant d'entrée est owners.

- Configurez les deuxième et troisième composants tELTMysqlInput de la même manière, mais sélectionnez cars et resellers respectivement comme nom de schéma.

Remarque :
Dans ce scénario, tous les schémas d’entrée sont stockés dans la zone **Metadata** du **Repository**, afin que vous les retrouviez facilement. Pour plus d’informations concernant les métadonnées, consultez le **Guide utilisateur du Studio Talend**.

Vous pouvez également sélectionner les trois composants d’entrée en déposant les schémas correspondants de la zone **Metadata** dans l’espace de modélisation graphique et en double-cliquant sur le composant **tELTMysqlInput** dans la boîte de dialogue [**Components**]. Cela vous permet d’éviter les étapes de nommage des composants et de définition de leur schéma.

- Reliez les trois composants **tELTMysqlInput** au composant **tELTMysqlMap** à l’aide de liens nommés strictement selon le nom des tables de la base de données : **owners**, **cars** et **resellers**.

- Reliez ensuite le composant **tELTMysqlMap** au **tELTMysqlOutput** et nommez le lien **agg_result**, qui est le nom de la table de la base de données dans laquelle vous allez sauvegarder le résultat de l’agrégation.

- Cliquez sur le composant **tELTMysqlMap** afin d’afficher sa vue **Basic settings** et configurer ses propriétés.

- Sélectionnez **Repository** dans la liste **Property Type** et sélectionnez la connexion à la base de données utilisée pour les composants d’entrée.
  
  Toutes les informations de la base de données sont automatiquement récupérées.

- Laissez les paramètres configurés par défaut

- Double-cliquez sur le composant **tELTMysqlMap** afin d’ouvrir le **Map Editor** pour mettre en place les jointures entre les tables d’entrée (Input) et configurer le flux de sortie.
Ajoutez les tables d’entrée en cliquant sur le bouton [+] dans le coin supérieur gauche de l’éditeur et sélectionnez les noms des tables correspondants dans la boîte de dialogue [Add a new alias].

Glissez-déposez la colonne ID_Owner à partir de la table owners vers la colonne correspondante dans la table cars.

Dans la table cars, cochez la case Explicit Join en face de ID_Owners. Une jointure INNER JOIN, le type de jointure par défaut, s’affiche dans la liste des jointures.

Glissez la colonne ID_Resellers à partir de la table cars vers la colonne correspondante de la table Resellers pour mettre en place la seconde jointure. Définissez à nouveau une option de jointure INNER JOIN.

Puis sélectionnez les colonnes à agréger dans la table de sortie, agg_result.

Déposez les colonnes ID_Owners, Name et ID_Insurance de la table owners dans la table de sortie.

Déposez les colonnes Registration, Make et Color de la table cars dans la table de sortie.

Déposez les colonnes Name_Reseller et City de la table resellers dans la table de sortie.

Avec les colonnes correspondantes sélectionnées, la mise en correspondance apparaît en jaune et les jointures en violet.

Appliquez un filtre sur la table de sortie. Cliquez sur le bouton Add filter row en haut de la table de sortie afin d’afficher le champ textuel Additional clauses, déposez la colonne City de la table resellers dans le champ textuel puis définissez la clause WHERE: resellers.City = 'Augusta'.

Une jointure INNER JOIN, le type de jointure par défaut, s’affiche dans la liste des jointures.
• Cliquez sur l’onglet **Generated SQL Select query** afin d’afficher l’instruction SQL correspondante.

```sql
SELECT
FROM
    owners INNER JOIN cars ON (cars.ID_owners = owners.ID_owner )
    INNER JOIN resellers ON (resellers.ID_Reseller = cars.ID_Reseller )
WHERE resellers.City = 'Augusta'
```

• Cliquez **OK** pour sauvegarder le paramétrage de l’ELT Map.

• Double-cliquez sur le composant **tELTMysqlOutput** afin d’afficher sa vue **Basic settings**.

• Dans la liste **Action on data** sélectionnez l’action que vous souhaitez effectuer sur les données.
• Sélectionnez **Repository** dans la liste **Schema** et définissez le schéma de sortie de la même manière que pour les schémas d’entrée. Notez que le schéma de sortie doit être nommé comme la table dans laquelle vous souhaitez sauvegarder vos résultats d’agrélation, **agg_result** dans ce scénario.

**Remarque** :
Vous pouvez également utiliser un schéma en mode **Built-In** et récupérer la structure du schéma du composant précédent. Cependant, vous devez être sûr d’avoir spécifié une table cible existant dans votre base de données, ayant la même structure de données.

• Laissez les autres paramètres configurés par défaut.

• Sauvegardez votre Job et appuyez sur la touche **F6** pour l’exécuter.

Toutes les données sélectionnées sont insérées dans la table **agg_result** comme spécifié dans l’instruction SQL.

<table>
<thead>
<tr>
<th>ID_Owner</th>
<th>Name</th>
<th>ID_Insurance</th>
<th>Registration</th>
<th>Make</th>
<th>Color</th>
<th>Name_Reseller</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td>John ROOSEVELT</td>
<td>04 UDR 821</td>
<td>04 UDR 821</td>
<td>Peugeot</td>
<td>green</td>
<td>BILL GARFIELD</td>
<td>AUGUSTA</td>
</tr>
<tr>
<td>11</td>
<td>Andrew COOLIDGE</td>
<td>104 ZYX 387</td>
<td>104 ZYX 387</td>
<td>EMW</td>
<td>yellow</td>
<td>Ulysses FOLK</td>
<td>AUGUSTA</td>
</tr>
<tr>
<td>10</td>
<td>Martin ADAMS</td>
<td>167 ZFF 318</td>
<td>167 ZFF 318</td>
<td>Audi</td>
<td>blue</td>
<td>Rutherford HARRISON</td>
<td>AUGUSTA</td>
</tr>
<tr>
<td>NULL</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

**Scénario 2 : ELT utilisant une table Alias**

Le scénario suivant décrit un Job mappant des informations provenant des deux tables d’entrée et d’une table Alias servant de table d’entrée virtuelle, vers une table de sortie. La table **employees** contient l’ID des employés, le numéro de leur service, leur nom et l’ID de leur responsable. Ces responsables sont également considérés comme des employés et par conséquent inclus dans la table **employees**. La table **dept** contient les informations de service des employés. La table Alias récupère le nom des responsables de la table **employees**.

• Glissez et déposez de la Palette dans l’espace de modélisation graphique deux composants **tELTMySqlInput**, un **tELTMySqlMap** et un **tELTMySqlOutput**.

Nommez-les comme vous le souhaitez afin de décrire au mieux leur fonctionnalité.

• Double-cliquez sur le premier composant **tELTMySqlInput** afin d’afficher sa vue **Basic settings** et configurer ses propriétés de base.
Sélectionnez Repository dans la liste Schema puis définissez la connexion à la base de données ainsi que le schéma, en cliquant sur le bouton [...] à côté du champ Edit Schema.

La connexion à la base de données est Talend_MySQL et le schéma du premier composant d’entrée est employees.

Remarque :
Dans ce scénario, les schémas d’entrée sont stockés dans le Repository, sous le nœud Metadata, afin que vous puissiez les récupérer plus facilement. Pour plus d’informations concernant les métadonnées, consultez le Guide utilisateur du Studio Talend.

Configurez le second composant tELTMySqlInput de la même manière, mais sélectionnez dept comme nom de schéma.

Double-cliquez sur le composant tELTMySqlOutput afin d’afficher sa vue Basic settings et définir ses propriétés.

Dans la liste Action on data, sélectionnez l’action que vous souhaitez effectuer, Insert dans ce scénario.

Dans la liste Schema, sélectionnez Repository et définissez le schéma de sortie de la même manière que pour les schémas d’entrée. Dans ce scénario, sélectionnez result comme schéma de sortie, qui est le nom de la table de la base de données utilisée pour stocker le résultat du mapping.

Le schéma de sortie contient toutes les colonnes des schémas d’entrée ainsi qu’une colonne ManagerName.

Laissez les autres paramètres configurés par défaut.

Reliez les deux composants tELTMySqlInput au tELTMySqlMap à l’aide de liens Link nommés strictement selon le nom des tables d’entrée, employees et dept dans ce scénario.

Reliez le composant tELTMySqlMap au tELTMySqlOutput à l’aide d’un lien Link. Cliquez sur Yes (Oui) à l’ouverture de la boîte de dialogue pour permettre au Mapper ELT de récupérer la structure de la table de sortie à partir du schéma de sortie.
• Cliquez sur le composant **tELTMysqlMap** afin d'afficher sa vue **Basic settings** et configurer ses propriétés de base.

Dans la liste **Property Type**, sélectionnez **Repository** puis sélectionnez la connexion à la base de données utilisée dans les composants d’entrée. Les informations de connexion à la base de données sont automatiquement récupérées.

• Laissez les autres paramètres configurés par défaut.

• Cliquez sur le bouton […] à côté du champ **ELT Mysql Map Editor** ou double-cliquez sur le composant **tELTMysqlMap** dans l’espace de modélisation graphique afin d’ouvrir le **ELT Map Editor**.

Le composant **tELTMysqlMap** est connecté au composant de sortie, la table de sortie est donc affichée dans la zone correspondante.

• Ajoutez les tables d’entrée employees et dept dans la zone d’entrée, en cliquant sur le bouton [*+] et en sélectionnant le nom des tables correspondant dans la boîte de dialogue [Add a new alias].

• Lorsque vous créez une table Alias à partir de la table employees en sélectionnant employees dans la liste **Select the table to use** et en saisissant Managers dans le champ **Type in a valid alias**, dans la boîte de dialogue [Add a new alias].

• Déposez la colonne DeptNo de la table employees dans la table dept.

• Cochez la case **Explicit join** devant la colonne DeptNo de la table dept afin de configurer une jointure **Inner Join**.

• Déposez la colonne ManagerId de la table employees dans la colonne ID de la table Managers.

• Cochez la case **Explicit join** devant la colonne ID de la table Managers et sélectionnez **LEFT OUTER JOIN** dans la list **Join**, afin que les lignes de sortie puissent contenir des valeurs Null.
• Déposez toutes les colonnes de la table *employees* dans les colonnes correspondantes de la table de sortie.

• Déposez les colonnes *DeptName* et *Location* de la table *dept* dans les colonnes correspondantes de la table de sortie.

• Déposez la colonne *Name* de la table *Managers* dans la colonne *ManagerName* de la table de sortie.

• Cliquez sur l’onglet *Generated SQL Select query* pour afficher l'instruction SQL à exécuter.

• Sauvegardez votre Job et appuyez sur F6 pour l’exécuter.
La table de sortie *result* de la base de données contient toutes les informations concernant les employés, y compris le nom de leurs responsables respectifs.

<table>
<thead>
<tr>
<th>ID</th>
<th>DeptNo</th>
<th>Name</th>
<th>ManagerID</th>
<th>DeptName</th>
<th>Location</th>
<th>ManagerName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>Alex</td>
<td>(NULL)</td>
<td>R&amp;D</td>
<td>New York</td>
<td>(NULL)</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>Peter</td>
<td>(NULL)</td>
<td>Accounting</td>
<td>Dallas</td>
<td>(NULL)</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>Mark</td>
<td>1</td>
<td>R&amp;D</td>
<td>New York</td>
<td>Alex</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>Michael</td>
<td>1</td>
<td>R&amp;D</td>
<td>New York</td>
<td>Alex</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>Christophe</td>
<td>2</td>
<td>Accounting</td>
<td>Dallas</td>
<td>Peter</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>Stephane</td>
<td>3</td>
<td>R&amp;D</td>
<td>New York</td>
<td>Mark</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>Cédric</td>
<td>3</td>
<td>R&amp;D</td>
<td>New York</td>
<td>Mark</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>Bill</td>
<td>4</td>
<td>R&amp;D</td>
<td>New York</td>
<td>Michael</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>Jack</td>
<td>2</td>
<td>Accounting</td>
<td>Dallas</td>
<td>Peter</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>Andrews</td>
<td>4</td>
<td>R&amp;D</td>
<td>New York</td>
<td>Michael</td>
</tr>
<tr>
<td>*</td>
<td>NULL</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>
tELTMysqlOutput

Ce composant exécute la requête Insert, Delete ou Update dans la base de données Mysql.

Les trois composants ELT MySQL sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données MySQL, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTMysqlOutput effectue l'action sur la table telle que spécifiée et insère les données selon le schéma de sortie défini dans le Mapper ELT.

Propriétés du tELTMysqlOutput Standard

Ces propriétés sont utilisées pour configurer le tELTMysqlOutput s'exécutant dans le framework de Jobs Standard.

Le composant tELTMysqlOutput Standard appartient à la famille ELT.
Le composant de ce framework est toujours disponible.

Basic settings

| Action on data | Sur les données de la table définie, vous pouvez effectuer les opérations suivantes :
|                | Insert : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s'arrête.
|                | Update : Met à jour des lignes existantes.
|                | Delete : Supprime les entrées correspondantes au flux d'entrée. |

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository.
|                       | Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|                       | • View schema : sélectionnez cette option afin de voir le schéma.
|                       | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
|                       | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |


**Where clauses for (for UPDATE and DELETE only)** Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.

**Default Table Name** Saisissez le nom de la table par défaut, entre guillemets doubles. Notez que la table doit déjà exister. Si elle n'existe pas, vous pouvez utiliser un composant `tCreateTable` pour en créer une d'abord. Pour plus d'informations concernant le `tCreateTable`, consultez `tCreateTable` à la page 575.

**Table name from connection name is variable** Cochez cette case lorsque le nom de la connexion à ce composant est configuré comme une variable, par exemple une variable de contexte.

**Use different table name** Cochez cette case afin de définir le nom d'une table de sortie différente, entre guillemets doubles, dans le champ `Table name` qui s'affiche.

**Mapping** Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d'informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types.

### Advanced settings

**tStatCatcher Statistics** Cochez la case afin de collecter les données de log au niveau de chaque composant.

### Global Variables

**Global Variables**

- **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable `After` et retourne un entier.
- **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable `After` et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case `Die on error` est décochée, si le composant a cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | L’utilisation du composant **tELTMySqlOutput** est étroitement liée à celle des composants **tELTMySqlInput** et **tELTMySqlMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

**Scénarios associés**

- **Scénario 1** : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- **Scénario 2** : ELT utilisant une table Alias à la page 794
- **Agrégier des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTNetezzaInput

Ce composant ajoute autant de tables que nécessaires dans une commande Insert qui peut être complexe.

Les trois composants ELT Netezza sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Netezza, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTNetezzaInput fournit le schéma de la table d’entrée à la commande SQL qui sera exécutée.

Propriétés du tELTNetezzaInput Standard

Ces propriétés sont utilisées pour configurer le tELTNetezzaInput s’exécutant dans le framework de Jobs Standard.

Le composant tELTNetezzaInput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit la nature le nombre de champs à traiter. Le schéma peut être Built-in ou distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|                        | • View schema : sélectionnez cette option afin de voir le schéma.
|                        | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
|                        | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
### Default Table Name

**Saisissez le nom de la table par défaut, entre guillemets doubles.**

### Advanced settings

**tStatCatcher Statistics**

**Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau du composant.**

### Global Variables

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

**Règle d’utilisation**

L’utilisation du composant **tELTNetezzaInput** est étroitement liée à celle des composants **tELTGreenplumMap** et **tELTGreenplumOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

### Scénarios associés

- **Scénario :** Mapper les données à l’aide d’une jointure implicite simple à la page 724
- **Scénario 1 :** Agréger les colonnes d’une table et appliquer un filtre à la page 790
- **Scénario 2 :** ELT utilisant une table Alias à la page 794
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
**tELTNetezzaMap**

Ce composant utilise les tables fournies en entrée pour alimenter les paramètres dans la commande mise en place. L'instruction peut inclure des jointures internes ou externes qui peuvent être implémentés entre des tables ou entre une table et ses alias.

Les trois composants ELT Netezza sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Netezza, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Le tELTNetezzaMap permet de construire votre commande SQL graphiquement en utilisant la table fournie en entrée.

**Propriétés du tELTNetezzaMap Standard**

Ces propriétés sont utilisées pour configurer le tELTNetezzaMap s’exécutant dans le framework de Jobs Standard.

Le composant tELTNetezzaMap Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà défini.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue <strong>Basic settings</strong> du composant de connexion créant cette connexion.</td>
</tr>
<tr>
<td></td>
<td>2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

| ELT Netezza Map Editor       | L’éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données. |
| Style link | Sélectionnez le type d’affichage des liens.  
**Auto** : par défaut, les liens entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de courbe.  
**Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne.  
**Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d’optimiser les performances. |
| --- | --- |
| Property type | Peut être **Built-in** ou **Repository**.  
**Built-in** : Propriétés utilisées ponctuellement.  
**Repository** : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |
| Host | Adresse IP du serveur de base de données. |
| Port | Numéro de port d’écoute du serveur de la base. |
| Database | Nom de la base de données. |
| Username et Password | Données d’authentification de l’utilisateur de la base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| Advanced settings |  
**Additional JDBC parameters** | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n’est pas disponible lorsque vous utilisez l’option **Use an existing connection** dans les **Basic settings**. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau du composant. |
| Global Variables |  
**Global Variables** | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. |
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | L’utilisation du composant tELTNetezzaMap est étroitement liée à celle des composants tELTNetezzaInput et tELTNetezzaOutput. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :** Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


### Scénarios associés

- **Scénario : Mapper les données à l’aide d’une jointure implicite simple** à la page 724
- **Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre** à la page 790
• Scénario 2 : ELT utilisant une table Alias à la page 794
• Scénario : Mapper des données à l’aide d’une sous-requête à la page 848, scénario utilisant une sous-requête
• Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion à la page 767
tELTNetezzaOutput

Ce composant exécute les actions Insert, Update ou Delete sur les données dans la table Netezza spécifiée, via une instruction SQL générée par le composant tELTNetezzaMap.

Les trois composants ELT Netezza sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Netezza, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Propriétés du tELTNetezzaOutput Standard

Ces propriétés sont utilisées pour configurer le tELTNetezzaOutput s'exécutant dans le framework de Jobs Standard.

Le composant tELTNetezzaOutput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Action on data | Sur les données de la table définie, vous pouvez effectuer les opérations suivantes :
|               | **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s’arrête.
|               | **Update** : Met à jour des lignes existantes.
|               | **Delete** : Supprime les entrées correspondantes au flux d’entrée. |

| Schema et Edit Schema | Un schéma est une description de lignes, il définit la nature le nombre de champs à traiter. Le schéma peut être **Built-in** ou distant dans le **Repository**.
|                       | Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
|                       | • **View schema** : sélectionnez cette option afin de voir le schéma.
|                       | • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
|                       | • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
|                       | Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent dans le Job. |
### Built-in
Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository

### Where clauses for (for UPDATE and DELETE only)
Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.

### Default Table Name
Saisissez le nom de la table par défaut, entre guillemets doubles.

### Table name from connection name is variable
Cochez cette case lorsque le nom de la connexion à ce composant est configuré comme une variable, par exemple une variable de contexte.

### Use different table name
Cochez cette case afin de définir le nom d'une table de sortie différente, entre guillemets doubles, dans le champ Table name qui s'affiche.

### Mapping
Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d'informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types.

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace</td>
</tr>
</tbody>
</table>

809
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d'utilisation | L'utilisation du composant `tELTNetezzaOutput` est étroitement liée à celle des composants `tELTNetezzaMap` et `tELTNetezzaInput`. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.  

> i Remarque :  

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma. |

### Scénarios associés

- Scénario : Mapper les données à l'aide d'une jointure implicite simple à la page 724
- Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre à la page 790
- Scénario 2 : ELT utilisant une table Alias à la page 794
- Agréger des données Snowflake à l'aide de variables de contexte comme noms de tables et de connexion à la page 767
tELTOracleInput

Ce composant fournit le schéma de la table Oracle à utiliser par le composant tELTOracleMap afin de générer l'instruction SQL SELECT.

Les trois composants ELT Oracle sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Oracle, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Propriétés du tELTOracleInput Standard

Ces propriétés sont utilisées pour configurer le tELTOracleInput s’exécutant dans le framework de Jobs Standard.

Le composant tELTOracleInput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l’utiliser dans la commande SQL Insert. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

* View schema : sélectionnez cette option afin de voir le schéma.
* Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
* Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Default Table Name**  
Saisissez le nom de la table par défaut, entre guillemets doubles.

**Default Schema Name**  
Saisissez le nom du schéma par défaut, entre guillemets doubles.

### Advanced settings

**tStatCatcher Statistics**  
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**  
L’utilisation du composant tELTOraclInput est étroitement liée à celle du tELTOOracleMap. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**
Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

### Scénarios associés

- **Scénario : Mettre à jour les lignes d’une base Oracle** à la page 816
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTOracleMap

Ce composant construit l'instruction SQL SELECT en utilisant les schémas des tables fournis par un ou plusieurs composant(s) tELTOracleInput.

Les trois composants ELT Oracle sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Oracle, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Propriétés du tELTOracleMap Standard

Ces propriétés sont utilisées pour configurer le tELTOracleMap s'exécutant dans le framework de Jobs Standard.

Le composant tELTOracleMap Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
</tr>
<tr>
<td></td>
<td>2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

| ELT Oracle Map Editor | L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données. |

| Style link | Sélectionnez le type d'affichage des liens. **Auto** : par défaut, les liens entre les schémas d'entrée et de sortie et les paramètres du service Web sont en forme de courbe. |
### Bezier curve
Les liens entre les schémas et les paramètres du service Web sont en forme de ligne.

### Line
Les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d’optimiser les performances.

**Property type**
Peut être Built-in ou Repository.

**Built-in**
Propriétés utilisées ponctuellement.

**Repository**
Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.

**Connection type**
Liste déroulante des pilotes disponibles.

**DB Version**
Sélectionnez la version d’Oracle que vous utilisez.

**Host**
Adresse IP du serveur de base de données.

**Port**
Numéro de port d’écoute du serveur de la base.

**Database**
Nom de la base de données.

**Username et Password**
Données d’authentification de l’utilisateur de la base. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

---

### Advanced settings

**Additional JDBC Parameters**
Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n’est pas disponible lorsque vous utilisez l’option Use an existing connection dans les Basic settings.

**Use Hint Options**
Cochez cette case afin d’activer la zone de configuration des hints, afin de vous permettre d’optimiser l’exécution d’une requête. Dans cette zone, les paramètres sont :

- **-HINT** : spécifiez le hint dont vous avez besoin, à l’aide de la syntaxe /*+ */.
- **-POSITION** : spécifiez où placer le hint dans l’instruction SQL.
- **-SQL STMT** : sélectionnez l’instruction SQL à utiliser.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau du composant.

---

**Global Variables**

**ERROR_MESSAGE**
message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est
une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

## Utilisation

### Règle d’utilisation

L’utilisation du composant *tELTOracleMap* est étroitement liée à celle des composants *tELTOracleInput* et *tELTOracleOutput*. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

### Dynamic settings

Cliquez sur le bouton [*+*] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table *Dynamic settings* est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue *Basic settings*. Lorsqu’un paramètre dynamique est configuré, la liste *Component List* de la vue *Basic settings* devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez *Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte* à la page 2641 et *Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement* à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*. 
Scénario : Mettre à jour les lignes d'une base Oracle

Le scénario suivant est basé sur le scénario d’agrégation, dans Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre à la page 790. L’action sur les données Update étant disponible en base Oracle, le scénario suivant décrit un Job qui met à jour des données particulières dans la table `agg_result`.

- Comme décrit dans Scénario 1 : Agréger les colonnes d'une table et appliquer un filtre à la page 790, créez un Job d’agrégation de données utilisant les composants ELT pour la base de données Oracle : `tELTOracleInput`, `tELTOracleMap` et `tELTOracleOutput`, puis exécutez le Job afin de sauvegarder le résultat de l’agrégation dans une table de base de données nommée `agg_result`.

  Remarque :
  
  Lorsque vous définissez des filtres dans l’éditeur ELT Map, notez que les chaînes de caractères sont sensibles à la casse dans la base de données Oracle.

- Lancez l’éditeur ELT Map et ajoutez une nouvelle table de sortie nommée `update_data`.

- Ajoutez une ligne de filtre à la table `update_data` pour établir une relation entre tables d’entrée et de sortie : `owners.ID_OWNER = agg_result.ID_OWNER`.

- Glissez la colonne `MAKE` de la table `cars` à la table `update_data`.

- Glissez la colonne `NAME_RESELLER` de la table `resellers` à la table `update_data`.

- Ajoutez un modèle entouré de guillemets simples, `A8` dans cet exemple, dans la colonne `MAKE` de la table `cars`, précédé de deux barres verticales.

- Ajoutez `Sold by`, entouré de guillemets simples devant la colonne `Name_Reseller` dans la table `resellers`, avec deux barres verticales au milieu.
Vérifiez la requête Select générée dans l’onglet **Generated SQL select query**.

- **Vérifiez la requête Select générée dans l’onglet** **Generated SQL select query**.

```sql
"SELECT CARS.MAKE || ' A8', 'Sold by' || RESELLERS.NAME_RESELLER
FROM OWNERS INNER JOIN CARS ON CARS.ID_OWNERS = OWNERS.ID_OWNERS
INNER JOIN RESELLERS ON RESELLERS.ID_RESELLER = CARS.ID_RESELLER
WHERE OWNERS.ID_OWNERS = agg_result.ID_OWNERS"
```

- Cliquez sur **OK** pour valider les modifications et fermer le mapper ELT.

- Désactivez le composant **tELTOracleOutput** nommé **Agg_Result** en cliquant-droit dessus et en sélectionnant **Deactivate Agg_Result** dans le menu contextuel.

- Déposez un nouveau composant **tELTOracleOutput** de la Palette dans l’espace de modélisation graphique. Nommez-le **Update_Data** afin d’identifier plus clairement sa fonction.

- Reliez le composant **tELTOracleMap** au nouveau **tELTOracleOutput** à l’aide du lien correspondant à la nouvelle table de sortie définie dans le mapper, **update_data** dans ce scénario.

- Double-cliquez sur le composant **tELTOracleOutput** afin d’afficher sa vue **Component** et définir ses propriétés.

- Sélectionnez **Update** dans la liste **Action on data**.

- Vérifiez le schéma et cliquez sur **Sync columns** pour récupérer la structure du schéma du composant précédent.

- Dans le champ **WHERE clauses**, entrez la clause suivante : `agg_result.MAKE = 'Audi'`, pour mettre à jour les données relatives à la marque Audi dans la table de la base de données **agg_result**.
**Scénario associé**

- **Scénario : Mettre à jour les lignes d'une base Oracle** à la page 816
- **Scénario : Mapper des données à l'aide d'une sous-requête** à la page 848, scénario utilisant une sous-requête
- **Agréger des données Snowflake à l'aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTOOracleOutput

Ce composant effectue l’action Insert, Update, Delete ou Merge sur les données de la table Oracle spécifiée, via l’instruction SQL générée par le composant tELTOOracleMap.

Les trois composants ELT Oracle sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Oracle, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Propriétés du tELTOOracleOutput Standard

Ces propriétés sont utilisées pour configurer le tELTOOracleOutput s’exécutant dans le framework de Jobs Standard.

Le composant tELTOOracleOutput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

<table>
<thead>
<tr>
<th>Basic settings</th>
</tr>
</thead>
</table>
| **Action on data** | Sur les données de la table définie, vous pouvez effectuer les opérations suivantes :
  
  **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s’arrête.
  
  **Update** : Met à jour des lignes existantes.
  
  **Delete** : Supprime les entrées correspondantes au flux d’entrée.
  
  **MERGE** : met à jour et/ou insère les données dans la table. Notez que les options disponibles pour les opérations de MERGE sont différentes de celles disponibles pour les opérations d’Insert, d’Update ou de Delete.

  **Remarque** :

  Les variables globales suivantes sont disponibles :

  - NB_LINE_INSERTED : Nombre de lignes insérées pendant l’opération d’Insert.
  - NB_LINE_UPDATED : Nombre de lignes mises à jour pendant l’opération d’Update.
  - NB_LINE_DELETED : Nombre de lignes supprimées pendant l’opération de Delete.
  - NB_LINE_MERGED : Nombre de lignes insérées ou mises à jour pendant l’opération de MERGE.

| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. |
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].


**Where clauses for (for UPDATE and DELETE only)**

Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.

**Use Merge Update** (pour **MERGE**)

Cochez cette case pour mettre à jour des données de votre table de sortie.

**Column** : Liste des colonnes du flux d’entrée.

**Update** : Cochez la case correspondant au nom de la colonne que vous souhaitez mettre à jour.

**Use Merge Update Where Clause** : Cochez cette case et saisissez la clause WHERE permettant de filtrer les données à mettre à jour, si nécessaire.

**Use Merge Update Delete Clause** : Cochez cette case et saisissez la clause WHERE permettant de filtrer les données à supprimer et mettre à jour, si nécessaire.

**Use Merge Insert** (pour **MERGE**)

Cochez cette case pour insérer des données dans la table.

**Column** : Liste des colonnes du flux d’entrée.

**Check All** : Cochez la case correspondant au nom de la colonne que vous souhaitez insérer.

**Use Merge Update Where Clause** : Cochez cette case et saisissez la clause WHERE permettant de filtrer les données à insérer.

**Default Table Name**

Saisissez le nom de la table par défaut.

**Default Schema Name**

Saisissez le nom du schéma Oracle par défaut.
### Table name from connection name is variable
Cochez cette case lorsque le nom de la connexion à ce composant est configuré comme une variable, par exemple une variable de contexte.

### Use different table name
Cochez cette case afin de définir le nom d’une table de sortie différente, entre guillemets doubles, dans le champ Table name qui s’affiche.

### Mapping
 Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d’informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types.

### Advanced settings

#### Use Hint Options
Cochez cette case pour activer la zone de configuration des hints lorsque vous souhaitez utiliser un hint pour optimiser l’exécution d’une requête. Dans cette zone, les paramètres sont :
- **HINT** : spécifiez l’hint dont vous avez besoin dans une instruction SQL.
- **POSITION** : spécifiez où mettre l’hint dans une instruction SQL.
- **SQL STMT** : sélectionnez l’instruction SQL dont vous avez besoin.

#### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

#### Global Variables

- **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.
- **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable *After* et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | L’utilisation du composant `tELTOracleOutput` est étroitement liée à celle des composants `tELTOracleInput` et `tELTOracleMap`. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

**Scénario : Utiliser la fonction MERGE d’Oracle pour mettre à jour et insérer simultanément des données**

Ce scénario décrit un Job vous permettant d’ajouter de nouvelles informations client et de mettre à jour les informations client existantes dans une table de base de données à l’aide de la commande `MERGE` d’Oracle.

**Relier les composants**

**Procédure**

1. De la Palette, glissez-déposez les composants suivants dans l’espace de modélisation graphique : `tELTOracleInput`, `tELTOracleMap` ainsi qu’un `tELTOracleOutput`.
2. Renommez le `tELTOracleInput` en `new_customer`, le `tELTOracleMap` en `ELT Mapper` et le `tELTOracleOutput` en `merge_data`.
3. Reliez le `tELTOracleInput` au `tELTOracleMap` à l’aide d’un lien `Row > New Output (table)`.
   Dans la boîte de dialogue qui s’ouvre, saisissez le nom de la table, `NEW_CUSTOMERS`. Ce nom doit correspondre au nom de la table de la base de données existante.
4. Reliez le `tELTOracleMap` au `tELTOracleOutput` à l’aide d’un lien `Row > New Output (table)`.
   Dans la boîte de dialogue qui s’ouvre, saisissez le nom de la table de la base de données, `customers_merge`, qui recevra les résultats de l’opération de `MERGE`.

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le `tELTOracleInput` afin d’afficher sa vue `Basic settings`. 
2. Dans la liste Schema, sélectionnez Repository, cliquez sur le bouton [...] à côté du champ Edit Schema puis sélectionnez votre connexion de base de données ainsi que le schéma souhaité dans la boîte de dialogue [Repository Content].

Le nom du schéma sélectionné apparaît automatiquement dans le champ Default Table Name.
Dans ce scénario, la connexion à la base de données est Talend_Oracle et le schéma est new_customers.

Remarque :

Dans ce scénario, le schéma d’entrée est stocké sous le nœud Metadata du Repository afin de le retrouver facilement. Pour plus d’informations concernant les métadonnées, consultez le Guide utilisateur du Studio Talend.

Vous pouvez également sélectionner le composant d’entrée en déposant le schéma correspondant de la zone Metadata dans l’espace de modélisation graphique et en double-cliquant sur le tELTOracleInput dans la boîte de dialogue [Components]. Cela vous permet d’éviter les étapes de nommage du composant d’entrée ainsi que la définition manuelle de son schéma.

3. Cliquez sur le composant tELTOracleMap afin d’afficher sa vue Basic settings.
4. Sélectionnez **Repository** dans la liste **Property Type** et choisissez la connexion à la base de données que vous avez utilisée pour les composants d’entrée.
Les informations relatives à la base de données sont automatiquement récupérées.
Laisssez les autres champs tels qu’ils sont.

5. Double-cliquez sur le composant **tELTOracleMap** afin d’ouvrir l’**ELT Map Editor** et configurer le flux de transformation de données.
Affichez la table d’entrée en cliquant sur le bouton [+ ] dans le coin supérieur gauche de l’**ELT Map Editor** et en sélectionnant le nom de la table correspondante dans le boîte de dialogue **Add a new alias**.
Dans ce scénario, la seule table d’entrée est **new_customers**.

6. Sélectionnez toutes les colonnes de la table d’entrée et déposez-les dans la table de sortie.

7. Cliquez sur **Generated SQL Select query** pour afficher l’instruction de la requête à exécuter.

Cliquez sur **OK** pour valider les paramètres **ELT Map** et fermer l’**ELT Map Editor**.
8. Dans l'espace de modélisation, double-cliquez sur le composant **tELTOracleOutput** pour afficher sa vue **Basic settings**.

Dans la liste **Action on data**, sélectionnez **MERGE**.

Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.

Cochez la case **Use Merge Update** pour mettre à jour les données via la fonction MERGE d'Oracle.

9. Dans le tableau qui apparaît, cochez les cases correspondant aux colonnes que vous souhaitez mettre à jour. L’objectif est de mettre à jour les données client en fonction de leur ID. Ainsi, cochez toutes les cases sauf celle correspondant à la colonne *ID*.

Avertissement :

Les colonnes définies comme clé primaire ne peuvent pas et ne doivent pas faire l’objet d’une mise à jour.

10. Cochez la case **Use Merge Insert** afin d’insérer de nouvelles données lors de la mise à jour des données existantes en utilisant la fonction MERGE d’Oracle.

Dans le tableau qui apparaît, cochez la case des colonnes dans lesquelles vous souhaitez insérer de nouvelles données.

Dans ce scénario, vous allez insérer de nouvelles données client. Cochez donc toutes les cases en cliquant sur la case **Check All**.

11. Renseignez le champ **Default Table Name** en saisissant le nom de la table cible déjà existante dans votre base de données. Dans cet exemple, saisissez *customers_merge*.

12. Laissez les autres paramètres tels qu’ils sont.
Exécuter le Job

Procédure
1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.

Les données sont mises à jour et insérées dans la base de données. La requête utilisée est affichée dans la console.

```
[statistics] connecting to socket on port 3712
[statistics] connected
Merge with:
MERGE INTO customers_merge target
USING (SELECT NEW_CUST_ID, NEW_CUST_FIRST_NAME, NEW_CUST_LAST_NAME, NEW_CUST_ADDRESS, NEW_CUST_REGISTERDATE, NEW_CUST_REVENUE, NEW_CUST_STATE FROM NEW_CUST WHERE)
WHEN MATCHED THEN UPDATE SET
target.FIRST_NAME = source.FIRST_NAME, target.LAST_NAME = source.LAST_NAME, target.ADDRESS = source.ADDRESS,
target.REVENUE = source.REVENUE, target.STATE = source.STATE
WHEN NOT MATCHED THEN INSERT (
ID, FIRST_NAME, LAST_NAME, REGISTERDATE, REVENUE, STATE)
VALUES (source.ID, source.FIRST_NAME, source.LAST_NAME, source.ADDRESS, source.REVENUE, source.STATE)
Merge with:
MERGE INTO customers_merge target
USING (SELECT NEW_CUST_ID, NEW_CUST_FIRST_NAME, NEW_CUST_LAST_NAME, NEW_CUST_ADDRESS, NEW_CUST_REGISTERDATE, NEW_CUST_REVENUE, NEW_CUST_STATE FROM NEW_CUST WHERE)
WHEN MATCHED THEN UPDATE SET
target.FIRST_NAME = source.FIRST_NAME, target.LAST_NAME = source.LAST_NAME, target.ADDRESS = source.ADDRESS,
target.REVENUE = source.REVENUE, target.STATE = source.STATE
WHEN NOT MATCHED THEN INSERT (
ID, FIRST_NAME, LAST_NAME, REGISTERDATE, REVENUE, STATE)
VALUES (source.ID, source.FIRST_NAME, source.LAST_NAME, source.ADDRESS, source.REVENUE, source.STATE)
[statistics] disconnected
```
tELTPostgresqlInput

Ce composant fournit le schéma de la table Postgresql à utiliser par le composant tELTPostgresqlMap afin de générer l’instruction SQL SELECT.

Les trois composants ELT Postgresql sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Postgresql, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Propriétés du tELTPostgresqlInput Standard

Ces propriétés sont utilisées pour configurer le tELTPostgresqlInput s’exécutant dans le framework de Jobs Standard.

Le composant tELTPostgresqlInput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l’utiliser dans la commande SQL Insert. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content. |


| Default Table Name | Saisissez le nom de la table par défaut. |
**Default Schema Name**

Saisissez le nom du schéma par défaut.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

**Règle d’utilisation**

L’utilisation du composant *tELTPostgresqlInput* est étroitement liée à celle du *tELTPostgresqlMap*. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque** :

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

**Scénarios associés**

- **Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre** à la page 790
- **Scénario 2 : ELT utilisant une table Alias** à la page 794
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTPostgresqlMap

Ce composant construit l'instruction SQL SELECT à l'aide des schémas de tables fournis par un ou plusieurs composant(s)tELTPostgresqlInput.

Les trois composants ELT Postgresql sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Postgresql, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Propriétés du tELTPostgresqlMap Standard

Ces propriétés sont utilisées pour configurer le tELTPostgresqlMap s'exécutant dans le framework de Jobs Standard.

Le composant tELTPostgresqlMap Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque :</strong> Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
</tr>
<tr>
<td></td>
<td>2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

| ELT Postgresql Map Editor | L'éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données. |

| Style link                | Sélectionnez le type d’affichage des liens. *Auto* : par défaut, les liens entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de courbe. |
Curves : les liens entre les schémas et les paramètres du service Web sont en forme de ligne.

Line : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d’optimiser les performances.

**Property type**

Peut être Built-in ou Repository.

**Built-in** : Propriétés utilisées ponctuellement.

**Repository** : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro de port d’écoute du serveur de la base.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>

**Username et Password**

Données d’authentification de l’utilisateur de la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Advanced settings**

**Additional JDBC parameters**

Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n’est pas disponible lorsque vous utilisez l’option **Use an existing connection** dans les **Basic settings**.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à L’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | L’utilisation du composant `tELTPostgresqlMap` est étroitement liée à celle des composants `tELTPostgresqlInput` et `tELTPostgresqlOutput`. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

> **Remarque :**

> Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

| Dynamic settings | Cliquez sur le bouton `[+]` pour ajouter une ligne à la table. Dans le champ `Code`, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table `Dynamic settings` est disponible uniquement lorsque la case `Use an existing connection` est cochée dans la vue `Basic settings`. Lorsqu’un paramètre dynamique est configuré, la liste `Component List` de la vue `Basic settings` devient inutilisable.


**Scénarios associés**

- **Scénario 1** : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- **Scénario 2** : ELT utilisant une table Alias à la page 794
- **Scénario** : Mapper des données à l’aide d’une sous-requête à la page 848, scénario utilisant une sous-requête
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTPostgresqlOutput

Ce composant effectue l’action Insert, Update ou Delete sur les données de la table Postgresql spécifiée, via l’instruction SQL générée par le composant tELTPostgresqlMap.

Les trois composants ELT Postgresql sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Postgresql, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Propriétés du tELTPostgresqlOutput Standard

Ces propriétés sont utilisées pour configurer le tELTPostgresqlOutput s’exécutant dans le framework de Jobs Standard.

Le composant tELTPostgresqlOutput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Action on data | Sur les données de la table définie, vous pouvez effectuer les opérations suivantes :
|                | **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s’arrête.
|                | **Update** : Met à jour des lignes existantes.
|                | **Delete** : Supprime les entrées correspondantes au flux d’entrée. |

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
|                        | • **View schema** : sélectionnez cette option afin de voir le schéma.
|                        | • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
|                        | • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |


<table>
<thead>
<tr>
<th>Where clauses for (for UPDATE and DELETE only)</th>
<th>Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Table Name</td>
<td>Saisissez le nom de la table par défaut.</td>
</tr>
<tr>
<td>Default Schema Name</td>
<td>Saisissez le nom du schéma par défaut.</td>
</tr>
<tr>
<td>Table name from connection name is variable</td>
<td>Cochez cette case lorsque le nom de la connexion à ce composant est configuré comme une variable, par exemple une variable de contexte.</td>
</tr>
<tr>
<td>Use different table name</td>
<td>Cochez cette case afin de définir le nom d’une table de sortie différente, entre guillemets doubles, dans le champ Table name qui s’affiche.</td>
</tr>
<tr>
<td>Mapping</td>
<td>Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d’informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier. ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |

833
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>L’utilisation du composant <code>tELTPostgresqlOutput</code> est étroitement liée à celle des composants <code>tELTPostgresqlInput</code> et <code>tELTPostgresqlMap</code>. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remarque</strong></td>
<td>Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.</td>
</tr>
</tbody>
</table>

**Scénarios associés**

- **Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre** à la page 790
- **Scénario 2 : ELT utilisant une table Alias** à la page 794
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTSybaseInput

Ce composant fournit le schéma de la table Sybase à utiliser par le composant tELTSybaseMap.

Les trois composants ELT Sybase sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Sybase, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Propriétés du tELTSybaseInput Standard

Ces propriétés sont utilisées pour configurer le tELTSybaseInput s'exécutant dans le framework de Jobs Standard.

Le composant tELTSybaseInput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Schema et Edit schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l'utiliser dans la commande SQL Insert. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles : • View schema : sélectionnez cette option afin de voir le schéma. • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales. • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Table Name</td>
<td>Saisissez le nom de la table par défaut.</td>
</tr>
</tbody>
</table>


**Default Schema Name**
Saisissez le nom du schéma Sybase par défaut.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | L’utilisation du composant tELTSybaseInput est étroitement liée à celle du tELTSybaseMap. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. Remarque : Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma. |
| Limitation | Ce composant requiert l’installation des fichiers .jar liés. |

**Scénarios associés**

- Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- Scénario 2 : ELT utilisant une table Alias à la page 794
- Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion à la page 767
tELTSybaseMap

Ce composant construit l'instruction SQL SELECT à l'aide des schémas de tables fournis par un ou plusieurs composant(s) tELTSybaseInput.

Les trois composants ELT Sybase sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Sybase, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Propriétés du tELTSybaseMap Standard

Ces propriétés sont utilisées pour configurer le tELTSybaseMap s'exécutant dans le framework de Jobs Standard.

Le composant tELTSybaseMap Standard appartient à la famille ELT.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Remarque :</td>
</tr>
<tr>
<td></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
</tr>
<tr>
<td></td>
<td>2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

| ELT Sybase Map Editor     | L’éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données. |

<table>
<thead>
<tr>
<th>Style link</th>
<th>Sélectionnez le type d’affichage des liens.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Auto : par défaut, les liens entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de courbe.</td>
</tr>
</tbody>
</table>
**Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne.

**Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d’optimiser les performances.

**Property type**

Peut être **Built-in** ou **Repository**.

**Built-in** : Propriétés utilisées ponctuellement.

**Repository** : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.

**DB Version**

Sélectionnez dans la liste déroulante la version de la base de données Sybase à utiliser.

**Host**

Adresse IP du serveur de base de données.

**Port**

Numéro de port d’écoute du serveur de la base de données.

**Database**

Nom de la base de données.

**Username et Password**

Données d’authentification de l’utilisateur de la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaine de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.
Utilisation

| Règle d’utilisation | L’utilisation du composant tELTSybaseMap est étroitement liée à celle des composants tELTSybaseInput et tELTSybaseOutput. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**
Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


| Limitation | Ce composant requiert l’installation des fichiers .jar liés.

Scénarios associés

- **Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre** à la page 790
- **Scénario 2 : ELT utilisant une table Alias** à la page 794
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
**tELTSybaseOutput**

Ce composant effectue l’action Insert, Update ou Delete sur les données de la table Sybase spécifiée via l’instruction SQL générée par le composant tELTSybaseMap.

Les trois composants ELT Sybase sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Sybase, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

**Propriétés du tELTSybaseOutput Standard**

Ces propriétés sont utilisées pour configurer le tELTSybaseOutput s’exécutant dans le framework de Jobs Standard.

Le composant tELTSybaseOutput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Action on data | Sur les données de la table définie, vous pouvez effectuer les opérations suivantes :
|               | Insert : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s’arrête.
|               | Update : Met à jour des lignes existantes.
|               | Delete : Supprime les entrées correspondantes au flux d’entrée. |

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository.
|                       | Le schéma défini est passé au Mapper ELT afin de l’insérer dans l’instruction SQL d’Insert.
|                       | Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|                       | • View schema : sélectionnez cette option afin de voir le schéma.
|                       | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
|                       | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
### Built-in
Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

### Repository
Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.

### Where clauses for (for UPDATE and DELETE only)
Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.

### Default Table Name
Saisissez le nom de la table par défaut.
Notez que la table doit déjà exister. Si elle n'existe pas, vous pouvez utiliser un composant `tCreateTable` pour en créer une d'abord. Pour plus d'informations concernant le `tCreateTable`, consultez `tCreateTable` à la page 575.

### Default Schema Name
Saisissez le nom du schéma Sybase par défaut.

### Table name from connection name is variable
Cochez cette case lorsque le nom de la connexion à ce composant est configuré comme une variable, par exemple une variable de contexte.

### Use different table name
Cochez cette case afin de définir le nom d’une table de sortie différente, entre guillemets doubles, dans le champ *Table name* qui s’affiche.

### Mapping
Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d’informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types.

### Advanced settings

#### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

#### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>Nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong></td>
<td>Nombre de lignes insérées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | L’utilisation du composant **tELTSybaseOutput** est étroitement liée à celle des composants **tELTSybaseInput** et **tELTSybaseMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

| Limitation | Ce composant requiert l’installation des fichiers .jar liés. |

### Scénarios associés

- **Scénario 1** : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- **Scénario 2** : ELT utilisant une table Alias à la page 794
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTTeradataInput

Ce composant fournit le schéma de la table Teradata à utiliser par le composant tELTTeradataMap afin de générer l’instruction SQL SELECT.

Les trois composants ELT Teradata sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Teradata, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

**Propriétés du tELTTeradataInput Standard**

Ces propriétés sont utilisées pour configurer le tELTTeradataInput s’exécutant dans le framework de Jobs Standard.

Le composant tELTTeradataInput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités. Le schéma est soit local (built-in) soit distant dans le Repository. Le schéma défini est ensuite passé au composant ELT Mapper afin de l’utiliser dans la commande SQL Insert.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Default Table Name**

Saisissez le nom par défaut de la table, entre guillemets doubles.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | L’utilisation du composant tELTTeradataInput est étroitement liée à celle du tELTTeradataMap. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table. **Remarque** : Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma. |

**Scénarios associés**

- **Scénario 1** : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- **Scénario 2** : ELT utilisant une table Alias à la page 794
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTTeradataMap

Ce composant construit l'instruction SQL SELECT à l'aide des schémas de tables fournis par un ou plusieurs composant(s) tELTTeradataInput.

Les trois composants ELT Teradata sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Teradata, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Propriétés du tELTTeradataMap Standard

Ces propriétés sont utilisées pour configurer le tELTTeradataMap s’exécutant dans le framework de Jobs Standard.

Le composant tELTTeradataMap Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

| ELT Teradata Map editor | L’éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.

| Property type | Peut être Built-in ou Repository.

| Built-in : Propriétés utilisées ponctuellement. |
**Repository** : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Données d'authentification de l'utilisateur de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

**Advanced settings**

**Query band**

Cochez cette case pour utiliser la fonctionnalité Teradata Query Banding afin d’ajouter les métadonnées à la requête à traiter, comme le nom de l’utilisateur exécutant la requête. Cela peut vous permettre, par exemple, d’identifier l’origine de la requête.

Une fois la case cochée, la table **Query Band parameters** s’affiche, dans laquelle vous pouvez saisir les informations des métadonnées à ajouter. Cette information prend la forme de paires clé/valeur, par exemple, **DpID** dans la colonne **Key** et **Finance** dans la colonne **Value**.


Cette case est indisponible lorsque vous avez coché la case **Using an existing connection**. Dans ce cas, si vous devez utiliser la fonctionnalité **Query band**, configurez-la dans l’onglet **Advanced settings** du composant de connexion Teradata à utiliser.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaine de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | L’utilisation du composant **tELTTeradataMap** est étroitement liée à celle des composants **tELTTeradataIn** et **tELTTeradataOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.  

---

| Utilisation | Règle d’utilisation | L’utilisation du composant **tELTTeradataMap** est étroitement liée à celle des composants **tELTTeradataIn** et **tELTTeradataOutput**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :**

Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.  

---
Scénario : Mapper des données à l'aide d'une sous-requête

Ce scénario décrit un Job mappant des données des deux tables d’entrée PreferredSubject et CourseScore vers la table de sortie TotalScoreOfPreferredSubject à l’aide d’une sous-requête.

La table PreferredSubject contient les données des matières préférées des étudiants. Pour reproduire ce scénario, vous pouvez charger des données dans la table à partir d’un fichier CSV comme suit. Pour plus d’informations concernant comment charger des données dans une table Teradata, consultez Scénario : Charger des données dans une base de données Teradata à la page 4152.

<table>
<thead>
<tr>
<th>SeqID</th>
<th>StuName</th>
<th>Subject</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Amanda</td>
<td>art</td>
<td>Amanda prefers art.</td>
</tr>
<tr>
<td>2</td>
<td>Ford</td>
<td>science</td>
<td>Ford prefers science.</td>
</tr>
<tr>
<td>3</td>
<td>Kate</td>
<td>art</td>
<td>Kate prefers art.</td>
</tr>
</tbody>
</table>

La table CourseScore contient les scores des étudiants dans leur matière favorite. Pour reproduire ce scénario, vous pouvez charger des données dans la table à partir d’un fichier CSV comme suit. Pour plus d’informations concernant comment charger des données dans une table Teradata, consultez Scénario : Charger des données dans une base de données Teradata à la page 4152.

<table>
<thead>
<tr>
<th>SeqID</th>
<th>StuName</th>
<th>Subject</th>
<th>Course</th>
<th>Score</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Amanda</td>
<td>science</td>
<td>math</td>
<td>85</td>
<td>science score</td>
</tr>
<tr>
<td>2</td>
<td>Amanda</td>
<td>science</td>
<td>physics</td>
<td>75</td>
<td>science score</td>
</tr>
<tr>
<td>3</td>
<td>Amanda</td>
<td>science</td>
<td>chemistry</td>
<td>80</td>
<td>science score</td>
</tr>
<tr>
<td>4</td>
<td>Amanda</td>
<td>art</td>
<td>chinese</td>
<td>85</td>
<td>art score</td>
</tr>
<tr>
<td>5</td>
<td>Amanda</td>
<td>art</td>
<td>history</td>
<td>95</td>
<td>art score</td>
</tr>
<tr>
<td>6</td>
<td>Amanda</td>
<td>art</td>
<td>geography</td>
<td>80</td>
<td>art score</td>
</tr>
<tr>
<td>7</td>
<td>Ford</td>
<td>science</td>
<td>math</td>
<td>95</td>
<td>science score</td>
</tr>
<tr>
<td>8</td>
<td>Ford</td>
<td>science</td>
<td>physics</td>
<td>85</td>
<td>science score</td>
</tr>
<tr>
<td>9</td>
<td>Ford</td>
<td>science</td>
<td>chemistry</td>
<td>80</td>
<td>science score</td>
</tr>
<tr>
<td>10</td>
<td>Ford</td>
<td>art</td>
<td>chinese</td>
<td>75</td>
<td>art score</td>
</tr>
<tr>
<td>11</td>
<td>Ford</td>
<td>art</td>
<td>history</td>
<td>80</td>
<td>art score</td>
</tr>
<tr>
<td>12</td>
<td>Ford</td>
<td>art</td>
<td>geography</td>
<td>85</td>
<td>art score</td>
</tr>
<tr>
<td>13</td>
<td>Kate</td>
<td>science</td>
<td>math</td>
<td>65</td>
<td>science score</td>
</tr>
<tr>
<td>14</td>
<td>Kate</td>
<td>science</td>
<td>physics</td>
<td>75</td>
<td>science score</td>
</tr>
<tr>
<td>15</td>
<td>Kate</td>
<td>science</td>
<td>chemistry</td>
<td>80</td>
<td>science score</td>
</tr>
<tr>
<td>16</td>
<td>Kate</td>
<td>art</td>
<td>chinese</td>
<td>85</td>
<td>art score</td>
</tr>
<tr>
<td>17</td>
<td>Kate</td>
<td>art</td>
<td>history</td>
<td>80</td>
<td>art score</td>
</tr>
<tr>
<td>18</td>
<td>Kate</td>
<td>art</td>
<td>geography</td>
<td>95</td>
<td>art score</td>
</tr>
</tbody>
</table>

Avant exécution du Job, il n’y a aucune donnée dans la table de sortie TotalScoreOfPreferredSubject : 

<table>
<thead>
<tr>
<th>SeqID</th>
<th>StuName</th>
<th>PreferredSubject</th>
<th>TotalScore</th>
</tr>
</thead>
</table>
Déposer et renommer les composants

Procédure

1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant à partir de la Palette : deux tELTTeradataInput, deux tELTTeradataMap et un tELTTeradataOutput.

2. Renommez les tELTTeradataInput en PreferredSubject et CourseScore respectivement, les deux tELTTeradataMap en ELTSubqueryMap et ELTMap et le tELTTeradataOutput en TotalScoreOfPreferredSubject.

Configurer les composants d’entrée

Procédure

1. Double-cliquez sur le composant PreferredSubject pour ouvrir sa vue Basic settings.

2. Dans le champ Default Table Name, saisissez un nom pour la table d’entrée. Dans cet exemple, nommez-la PreferredSubject.

3. Cliquez sur le bouton [...] à côté du champ Edit schema pour définir le schéma de la table d’entrée dans l’éditeur du schéma.

4. Connectez le PreferredSubject au ELTMap à l’aide d’un lien Link > PreferredSubject (Table).
5. Double-cliquez sur le CourseScore pour ouvrir sa vue **Basic settings**.

6. Dans le champ **Default Table Name**, saisissez le nom de la table d’entrée. Dans cet exemple, saisissez *CourseScore*.

7. Cliquez sur le bouton […] à côté du champ **Edit schema** pour définir le schéma de la table d’entrée dans l’éditeur du schéma.

Cliquez six fois sur le bouton [+] pour ajouter six colonnes, *SeqID* et *Score* dont le **DB Type** est **INTEGER**, *StuName*, *Subject*, *Course* et *Detail* dont le **DB Type** est **VARCHAR**.
Cliquez sur **OK** pour valider ces modifications et fermer l’éditeur du schéma.

8. Reliez le CourseScore au ELTSubqueryMap à l’aide d’un lien **Link > CourseScore (Table)**.

**Configurer le composant de sortie**

**Procédure**

1. Double-cliquez sur le composant TotalScoreOfPreferredSubject pour ouvrir sa vue **Basic settings**.
2. Dans le champ **Default Table Name**, saisissez un nom pour la table de sortie. Dans cet exemple, saisissez **TotalScoreOfPreferredSubject**.

3. Cliquez sur le bouton […] à côté du champ **Edit schema** afin de définir le schéma de la table de sortie dans l’éditeur du schéma.

Cliquez quatre fois sur le bouton [+], pour ajouter quatre colonnes, **SeqID** et **TotalScore** dont le **DB Type** est **INTEGER**, **StuName** et **PreferredSubject** dont le **DB Type** est **VARCHAR**.

Cliquez sur **OK** pour valider ces modifications et fermer l’éditeur du schéma.

**Configurer le mapping des données afin de générer une sous-requête**

**Procédure**

1. Double-cliquez sur le **ELTSubqueryMap** pour ouvrir sa vue **Basic settings**.
Vous n’avez pas besoin de spécifier les informations de connexion à la base de données Teradata dans le composant **ELTSubqueryMap**. Les informations de connexion sont spécifiées dans le composant **ELTMap**.

2. Cliquez sur le bouton [...] du champ **ELT Teradata Map Editor** pour ouvrir son éditeur.

3. Ajoutez la table d’entrée **CourseScore** en cliquant sur le bouton [+ ] dans le coin supérieur gauche de l’éditeur de mapping et sélectionnez le nom de la table dans la liste déroulante de la boîte de dialogue qui s’ouvre.

4. Ajoutez une table de sortie en cliquant sur le bouton [+ ] dans le coin supérieur droit de l’éditeur et en saisissant le nom de la table **TotalScore** dans le champ correspondant de la boîte de dialogue.

5. Cliquez sur les colonnes **StuName**, **Subject** et **Score** dans la table d’entrée et déposez-les dans la table de sortie.

6. Cliquez sur le bouton **Add filter row** dans le coin supérieur droit de la table de sortie et sélectionnez **Add an other(GROUP...) clause** dans le menu contextuel. Dans le champ **Additional other clauses (GROUP/ORDER BY...)**, saisissez la clause **GROUP BY CourseScore.StuName, CourseScore.Subject**.

   Ajoutez la fonction d’agrégation **SUM** pour la colonne **Score** de la table de sortie en modifiant l’expression de cette colonne en **SUM(CourseScore.Score)**.

7. Cliquez sur l’onglet **Generated SQL Select query for "table1" output** au bas de l’éditeur de mapping pour afficher l’instruction SQL correspondante générée.

Cette requête SQL apparaît comme sous-requête SQL générée par le composant **ELTMap**.
8. Cliquez sur OK pour valider les modifications et fermer l’éditeur de mapping.

9. Reliez le ELTSubqueryMap au ELTMap à l’aide d’un lien Link > TotalScore (table1). Notez que le lien est automatiquement renommé en TotalScore (Table_ref) car la table de sortie TotalScore est une table de référence.

**Mapper les schémas d’entrée et de sortie**

**Procédure**

1. Cliquez-droit sur le composant ELTMap, sélectionnez Link > *New Output* (Table) dans le menu contextuel et cliquez sur TotalScoreOfPreferredSubject. Dans la boîte de dialogue qui s’ouvre, cliquez sur Yes pour récupérer le schéma du composant cible.

2. Cliquez sur le ELTMap pour ouvrir sa vue Basic settings.

   ![Image](image.png)

   Dans les champs Host, Database, Username et Password, saisissez vos informations de connexion à la base de données Teradata.

3. Cliquez sur le bouton [...] à côté du champ ELT Teradata Map Editor pour ouvrir son éditeur.
4. Ajoutez la table d’entrée `PreferredSubject` en cliquant sur le bouton `[+]` dans le coin supérieur gauche de l’éditeur et sélectionnez le nom de la table souhaitée dans la liste de la boîte de dialogue.

   Répétez l’opération pour ajouter une autre table d’entrée, `TotalScore`.

5. Cliquez sur la colonne `StuName` de la table d’entrée `PreferredSubject` et déposez-la dans la colonne correspondante de la table d’entrée `TotalScore`. Cochez la case `Explicit join` de la colonne `StuName` dans la table `TotalScore`.

   Répétez l’opération avec la colonne `Subject`.

6. Cliquez sur la colonne `SeqID` de la table d’entrée `PreferredSubject` et déposez-la dans la colonne correspondante de la table de sortie.

   Répétez l’opération avec les colonnes `StuName` et `Subject` de la table d’entrée `PreferredSubject` et la colonne `Score` de la table d’entrée `TotalScore`.

7. Cliquez sur l’onglet `Generated SQL Select query for "table2" output` au bas de l’éditeur de mapping pour afficher l’instruction SQL correspondante générée.
La requête SQL générée dans le composant ELTSubqueryMap apparaît comme sous-requête dans la requête QSL générée par ce composant. L'alias est automatiquement ajouté pour les colonnes sélectionnées dans la sous-requête.

8. Cliquez sur OK pour valider les modifications et fermer l'éditeur de mapping.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour l'exécuter.

```
Starting job 202533_TELTeradataMap at 16:29 17/10/2014

[statistics] connecting to socket on port 3751
[statistics] connected
Inserting with :
INSERT INTO TotalScoreOfPreferredSubject
    (SegID, StuName, PreferredSubject, TotalScore) SELECT
    PreferredSubject, SegID, PreferredSubject.StuName, PreferredSubject.Subject, TotalScore.Score
    FROM PreferredSubject
    INNER JOIN ( SELECT CourseScore.StuName AS StuName, CourseScore.Subject AS Subject, SUM(CourseScore.Score) AS Score
        FROM CourseScore GROUP BY CourseScore.StuName, CourseScore.Subject ) TotalScore
    ON ( TotalScore.StuName EQ PreferredSubject.StuName AND TotalScore.Subject EQ PreferredSubject.Subject )
-- 0 rows inserted
```

```
Job 202533_TELTeradataMap ended at 16:29 17/10/2014 [exit code=0]
```

L'instruction sélectionnée est générée et les données de mapping sont écrites dans la table de sortie.

**Scénarios associés**

- **Scénario 1** : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- **Scénario 2** : ELT utilisant une table Alias à la page 794
- **Agréger des données Snowflake à l'aide de variables de contexte comme noms de tables et de connexion** à la page 767
tELTTeradataOutput

Ce composant effectue l’action Insert, Update ou Delete sur les données de la table Teradata spécifiée via l’instruction SQL générée par le composant tELTTeradataMap.

Les trois composants ELT Teradata sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Teradata, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

Propriétés du tELTTeradataOutput Standard

Ces propriétés sont utilisées pour configurer le tELTTeradataOutput s’exécutant dans le framework de Jobs Standard.

Le composant tELTTeradataOutput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Action on data | Sur les données de la table définie, vous pouvez effectuer les opérations suivantes :
|               | **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s’arrête.
|               | **Update** : Met à jour des lignes existantes.
|               | **Delete** : Supprime les entrées correspondantes au flux d’entrée. |

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository
|                      | Le schéma défini est passé au Mapper ELT afin de l’insérer dans l’instruction SQL d’Insert.
|                      | Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|                      | • **View schema** : sélectionnez cette option afin de voir le schéma.
|                      | • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
|                      | • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| **Built-in** | Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| **Where clauses for (for UPDATE and DELETE only)** | Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression. |
| **Default Table Name** | Saisissez le nom par défaut de la table, entre guillemets doubles. Noter que la table doit déjà exister. Si elle n'existe pas, vous pouvez utiliser un composant tCreateTable pour en créer une d'abord. Pour plus d’informations concernant le tCreateTable, consultez tCreateTable à la page 575. |
| **Table name from connection name is variable** | Cochez cette case lorsque le nom de la connexion à ce composant est configuré comme une variable, par exemple une variable de contexte. |
| **Use different table name** | Cochez cette case afin de définir le nom d’une table de sortie différente, entre guillemets doubles, dans le champ Table name qui s’affiche. |
| **Mapping** | Sélectionnez dans la liste le fichier de mapping des métadonnées pour la base de données à utiliser. Ce fichier est utilisé pour la conversion de type des données entre la base de données et Java. Pour plus d’informations concernant le mapping de métadonnées, consultez la documentation relative au Mapping de types. |

**Advanced settings**

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| **Global Variables** | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.  
**NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. |
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

| Règle d’utilisation | L’utilisation du composant **tELTTeradataOutput** est étroitement liée à celle des composants **tELTTeradataInput** et **tELTTeradataMap**. Notez que le lien de sortie (output) à utiliser avec ces deux composants doit respecter strictement la syntaxe du nom de la table.

**Remarque :** Les composants ELT ne traitent pas le flux de données lui-même mais uniquement les informations du schéma.

| Limitation | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

### Scénarios associés

- **Scénario 1** : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- **Scénario 2** : ELT utilisant une table Alias à la page 794
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
**tELTVerticaInput**

Ce composant fournit le schéma de la table Vertica à utiliser par le composant tELTVerticaMap afin de générer l'instruction SQL SELECT.

Les trois composants ELT Vertica sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Vertica, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

**Propriétés du tELTVerticaInput Standard**

Ces propriétés sont utilisées pour configurer le tELTVerticaInput s'exécutant dans le framework de Jobs Standard.

Le composant tELTVerticaInput Standard appartient à la famille ELT.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Schema et Edit schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <code>line</code> lors du nommage des champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-In</strong></td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <strong>No</strong> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td>Default Table Name</td>
<td>Saisissez le nom par défaut de la table.</td>
</tr>
</tbody>
</table>
**Default Schema Name**

Saisissez le nom par défaut du schéma.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

**Global Variables**

**Global Variables**

| ERROR_MESSAGE | message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

**Règle d’utilisation**

Le tELTVerticaInput est utilisé avec d’autres composants, notamment le tELTVerticaMap. Notez que le lien de sortie à utiliser avec ces composants doit correspondre à la syntaxe du nom de la table.

**Remarque :**

Les composants ELT ne gèrent pas de flux de données, uniquement des informations de schéma.

**Scénarios associés**

- **Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre** à la page 790
- **Scénario 2 : ELT utilisant une table Alias** à la page 794
- **Scénario : Mapper des données à l’aide d’une sous-requête** à la page 848
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
**tELTVerticaMap**

Ce composant construit l'instruction SQL SELECT en utilisant les schémas de tables fournis par un ou plusieurs composant(s) tELTVerticaInput.

Les trois composants ELT Vertica sont étroitement liés en ce qui concerne leurs conditions d'utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Vertica, afin de générier des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

**Propriétés du tELTVerticaMap Standard**

Ces propriétés sont utilisées pour configurer le tELTVerticaMap s'exécutant dans le framework de Jobs Standard.

Le composant tELTVerticaMap Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th><strong>DB Version</strong></th>
<th>Sélectionnez la version de la base de données Vertica à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td><strong>1.</strong></td>
<td>Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
</tr>
<tr>
<td><strong>2.</strong></td>
<td>Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>ELT Vertica Map Editor</strong></td>
<td>L’éditeur ELT du Mapper vous permet de définir le schéma de sortie ainsi que de construire graphiquement la commande SQL à exécuter. Le nom des colonnes du schéma peut être différent du nom des colonnes dans la base de données.</td>
</tr>
</tbody>
</table>
### Style link

Sélectionnez le type d’affichage des liens.

- **Auto** : par défaut, les liens entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de courbe.
- **Bezier curve** : les liens entre les schémas et les paramètres du service Web sont en forme de ligne.
- **Line** : les liens entre les schémas et les paramètres du service Web sont en forme de lignes droites. Cette option vous permettra d’optimiser les performances.

### Property Type

Peut être **Built-In** ou **Repository**.

- **Built-in** : Propriétés utilisées ponctuellement.

### Host

Saisissez l’adresse IP ou le nom de l’hôte de la base de données.

### Port

Saisissez le numéro du port d’écoute de la base de données.

### Database

Saisissez le nom de la base de données à utiliser.

### Additional JDBC Parameters

Spécifiez les propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez.

### Username et Password

Saisissez les informations d’authentification de l’utilisateur de la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

### Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

### Global Variables

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.
tELTVerticaMap

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

| Règle d'utilisation | Le tELTVerticaMap est utilisé avec les composants tELTVerticalInput et tELTVerticaOutput. Notez que le lien de sortie à utiliser avec ces composants doit correspondre à la syntaxe du nom de la table.

Remarque :

Les composants ELT ne gèrent pas de flux de données, seulement des informations relatives aux schémas. |

**Scénarios associés**

- **Scénario 1** : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- **Scénario 2** : ELT utilisant une table Alias à la page 794
- **Scénario** : Mapper des données à l’aide d’une sous-requête à la page 848, scénario utilisant une sous-requête
- **Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion** à la page 767
**tELTVerticaOutput**

Ce composant effectue l’action Insert, Update ou Delete sur les données de la table Vertica spécifiée via l’instruction SQL générée par le composant tELTVerticaMap.

Les trois composants ELT Vertica sont étroitement liés en ce qui concerne leurs conditions d’utilisation. Ces composants sont conçus pour prendre en charge les schémas de la base de données Vertica, afin de générer des instructions SQL, dont des clauses, qui sont exécutées dans la table de sortie définie.

**Propriétés du tELTVerticaOutput Standard**

Ces propriétés sont utilisées pour configurer le tELTVerticaOutput s’exécutant dans le framework de Jobs Standard.

Le composant tELTVerticaOutput Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Action on data | Sur les données de la table définie, vous pouvez effectuer les opérations suivantes :
|                | **Insert** : Ajoute de nouvelles lignes à la table. Si des doublons sont identifiés, le Job s’arrête.
|                | **Update** : Met à jour des lignes existantes.
|                | **Delete** : Supprime les entrées correspondantes au flux d’entrée. |

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs. |
|                       | **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Sync columns
Cliquez sur ce bouton pour récupérer le schéma du composant précédent connecté dans le Job.

### Where clauses (for UPDATE and DELETE only)
Saisissez la clause permettant de filtrer les données à mettre à jour ou à supprimer lors des opérations de mise à jour ou de suppression.

Ce champ est disponible uniquement lorsque l’action **Update** ou **Delete** est sélectionnée dans la liste **Action on data**.

### Default Table Name
Saisissez le nom par défaut de la table.

### Default Schema Name
Saisissez le nom par défaut du schéma.

### Table name from connection name is variable
Cochez cette case lorsque le nom de la connexion à ce composant est configuré comme une variable, par exemple une variable de contexte.

### Use different table name
Cochez cette case pour utiliser un nom différent pour la table de sortie.

### Table name
Saisissez le nom de la table de sortie.

Ce champ est disponible uniquement lorsque la case **Use different table name** est cochée.

### Advanced settings

#### Direct
Cochez cette case pour écrire les données directement dans le disque sans passer par la mémoire.

Cette case n’est pas visible lorsque la case **Set SQL Label** est cochée.

#### Set SQL Label
Cochez cette case et spécifiez le libellé identifiant la requête. Pour plus d’informations, consultez [How to label queries for profiling](en anglais). Cette case n’est pas visible lorsque la case **Direct** est cochée.

#### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</th>
</tr>
</thead>
</table>
NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d'utilisation

Le tELTVerticaOutput est utilisé avec d’autres composants, notamment le tELTVerticaMap. Notez que le lien de sortie à utiliser avec ces composants doit correspondre à la syntaxe du nom de la table.

Remarque :
Les composants ELT ne gèrent pas de flux de données mais des informations relatives au schéma.

Scénarios associés

- Scénario 1 : Agréger les colonnes d’une table et appliquer un filtre à la page 790
- Scénario 2 : ELT utilisant une table Alias à la page 794
- Scénario : Mapper des données à l’aide d’une sous-requête à la page 848
- Agréger des données Snowflake à l’aide de variables de contexte comme noms de tables et de connexion à la page 767
tESBConsumer

Ce composant appelle la méthode définie du Service Web invoqué et retourne la classe, selon les paramètres spécifiés.

**Propriétés du tESBConsumer Standard**

Ces propriétés sont utilisées pour configurer le tESBConsumer s'exécutant dans le framework de Jobs Standard.

Le composant tESBConsumer Standard appartient à la famille ESB.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Service Configuration</th>
<th>Description des bindings et de la configuration du Service Web. Le champ Endpoint est automatiquement rempli après complétion de la configuration du service.</th>
</tr>
</thead>
</table>

| Input schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur Sync columns afin de récupérer automatiquement les colonnes du composant précédent. |
|-----------------------------|--------------------------------------------------------------------------------------------------|

|---------|--------------------------------------------------------------------------------------------------|

|-----------|--------------------------------------------------------------------------------------------------|
### Response schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

---

### Built-in

Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

---

### Repository


---

### Fault Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
<table>
<thead>
<tr>
<th><strong>Built-in</strong></th>
<th>Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Service Locator</strong></td>
<td>Maintient la disponibilité du service pour répondre aux demandes et aux accords de niveau de service (Service Level Agreements). Cette option est indisponible lorsque la case Use Service Registry est cochée.</td>
</tr>
<tr>
<td><strong>Use Service Activity Monitor</strong></td>
<td>Capture les événements et stocke les informations pour permettre des analyses en profondeur de l’activité du service et un suivi des messages à travers une transaction métier. Cette option peut être utilisée, entre autres, pour analyser le temps de réponse du service,</td>
</tr>
</tbody>
</table>
identifier les modèles de trafic ou effectuer une analyse de cause racine.

Cette option est désactivée lorsque la case Use Service Registry est cochée, si vous avez souscrit à l'une des solutions Enterprise ESB de Talend.

| **Use Authentication** | Cochez cette case pour activer l'option d'authentification. Choisissez entre Basic HTTP, HTTP Digest, Username Token et SAML Token (ESB runtime only).
Saisissez un identifiant et un mot de passe dans les champs correspondants. Les options Basic HTTP, HTTP Digest et Username Token fonctionnent dans le studio et dans le conteneur d'exécution. L'authentification via SAML Token fonctionne uniquement dans le conteneur d'exécution.

Lorsque l’option SAML Token (ESB runtime only) est sélectionnée, vous pouvez fournir les informations d’authentification pour envoyer la requête ou effectuer l’appel si un utilisateur est authentifié, en propagant les informations d’authentification. Sélectionnez :
- : Saisissez l’identifiant et le mot de passe dans les champs correspondants afin d’accéder au service.

**Propagate using U/P** : Saisissez l’identifiant et le mot de passe utilisés pour vous authentifier via STS.

**Propagate using Certificate** : Saisissez l’alias et le mot de passe utilisés pour vous authentifier via STS.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

Cette option est indisponible lorsque la case Use Service Registry est cochée.

| **Use Business Correlation** | Cochez cette case pour créer un ID de corrélation dans ce composant.
Vous pouvez spécifier un ID de corrélation dans le champ Correlation Value. Dans ce cas, l'ID de corrélation est passé au service qu'il appelle, afin que les appels en chaîne de services soient groupés sous cet ID de corrélation. Si vous laissez ce champ vide, cette valeur est générée automatiquement au moment de l'exécution.

Lorsque cette option est activée, le tESBConsumer extrait également l'ID de corrélation de l'en-tête de réponse et le stocke dans le composant pour usage ultérieur dans le flux.

Cette option est automatiquement activée lorsque la case Use Service Registry est cochée.

| **Use GZip Compress** | Cochez cette case afin de compresser les messages entrants au format GZip avant envoi. |
### Die on error

| Cochez cette case si vous souhaitez arrêter le Job en cas d’erreur. |

### Advanced settings

#### Log messages

| Cochez cette case pour enregistrer l’échange de messages entre le service et le consommateur. |

#### Service Locator Custom Properties

| Cette table apparaît lorsque la case Use Service Locator est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ Property Name et Property Value respectivement. |

#### Service Activity Custom Properties

| Cette table apparaît lorsque la case Use Service Activity Monitor est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ Property Name et Property Value respectivement. |

#### Connection time out (second)

| Configurez une valeur en secondes avant interruption de la connexion au Service Web. Cette option fonctionne uniquement dans le studio. Pour l’utiliser après le déploiement du composant dans le moteur d’exécution :

1. Créez un fichier de configuration avec le nom org.apache.cxf.http.conduits-<endpoint_name>.cfg dans le dossier <TalendRuntimePath>/container/etc/.

2. Spécifiez l'url du service Web et le paramètre client.ConnectionTimeout de délai avant suspension en millisecondes dans le fichier de configuration. Si vous devez utiliser l’option Receive time out, spécifiez le délai client.Receive Timeout en millisecondes également. L'url peut être un endpoint ou une expression régulière contenant des caractères de remplacement, par exemple :

```python
url = http://localhost:8040/*
client.ConnectionTimeout=1000000
client.ReceiveTimeout=2000000
```


#### Receive time out (second)

| Configurez une valeur en secondes pour la réception de la réponse. Cette option fonctionne uniquement dans le studio. Pour l’utiliser après le déploiement du composant dans le |
moteur d'exécution, consultez l'option **Connection time out**.

### Disable Chunking

Cochez cette case pour désactiver l’encodage du payload par morceaux. En général, l’encodage par morceaux est plus efficace car il permet une mise en flux directe. Parfois, avec l’encodage par morceau, les payloads sont tronqués. Si vous obtenez des erreurs inhabituelles lorsque vous essayez d’utiliser un service, essayez de désactiver l’encodage par morceaux pour voir s’il y a une amélioration.

### Trust Server with SSL/TrustStore file et TrustStore password

Cochez cette case pour authentifier le serveur auprès du client via un protocole sécurisé SSL et renseignez les champs suivants :

- **TrustStore file** : saisissez le chemin d’accès et le nom du fichier TrustStore contenant la liste des certificats approuvés par le client.
- **TrustStore password** : saisissez le mot de passe utilisé pour vérifier l’intégrité des données TrustStore.

### Use http Proxy/Proxy host, Proxy port, Proxy user et Proxy password

Cochez cette case si vous utilisez un serveur proxy et renseignez les informations nécessaires. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

### HTTP Headers

Cliquez sur [+] autant de fois que nécessaire pour ajouter la ou les paires nom-valeurs pour les en-têtes HTTP afin de définir les paramètres de l’opération HTTP requise.

### tStatCatcher Statistics

Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CORRELATION_ID : l’ID de corrélation par lequel regr ouper les appels de services en chaîne. Cette variable est une variable <strong>Flow</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>HTTP_RESPONSE_CODE : code HTTP de statut. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>HTTP_HEADERS : ensemble des en-têtes HTTP de la réponse. Cette variable est une variable <strong>Flow</strong> et retourne un objet Map java.util.Map&lt;String,</td>
</tr>
</tbody>
</table>
Le nom de l’en-tête est représenté par la clé Map. Les valeurs des en-têtes sont représentées par `java.util.List<?>`.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé comme composant intermédiaire. Il nécessite un composant d’entrée et un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table et renseignez le champ <strong>Code</strong> avec une variable de contexte, afin d’activer ou désactiver dynamiquement l’option <strong>Use Authentication</strong> ou <strong>Use HTTP proxy</strong> lors de l’exécution. Vous pouvez ajouter deux lignes à la table afin de configurer les deux options. Une fois qu’un paramètre dynamique est défini, l’option correspondante est mise en relief et devient inutilisable dans les vues <strong>Basic settings</strong> et <strong>Advanced settings</strong>. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez <em>Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte</em> à la page 2641 et <em>Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement</em> à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
<tr>
<td><strong>Limitation</strong></td>
<td>Une JDK est requise pour que ce composant puisse fonctionner.</td>
</tr>
</tbody>
</table>

**Scénario 1 : Utiliser le composant tESBConsumer pour récupérer une adresse e-mail valide**

Ce scénario décrit un Job utilisant un composant **tESBConsumer** pour récupérer une adresse e-mail valide.
**Construire le Job**

**Procédure**

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un tFixedFlowInput, un tESBConsumer, deux tXMLMap et deux tLogRow.

2. Cliquez-droit sur le composant tFixedFlowInput, sélectionnez Row > Main dans le menu contextuel et cliquez sur le premier tXMLMap.

3. Cliquez-droit sur le composant tXMLMap, sélectionnez Row > *New Output* (Main) dans le menu contextuel et cliquez sur le tESBConsumer. Saisissez payload dans la boîte de dialogue afin de nommer ce lien et acceptez la propagation afin de récupérer le schéma du composant tESBConsumer.

4. Cliquez-droit sur le tESBConsumer, sélectionnez Row > Response dans le menu contextuel et cliquez sur le second tLogRow.

5. Cliquez-droit sur le second tXMLMap, sélectionnez Row > *New Output* (Main) dans le menu contextuel et cliquez sur le second tLogRow. Saisissez response dans la boîte de dialogue qui s’ouvre afin de nommer le lien.

6. Cliquez-droit sur le composant tESBConsumer, sélectionnez Row > Fault dans le menu contextuel et cliquez sur le second tLogRow.

**Configurer les composants**

Les tLogRow monitorent les échanges de messages de réponses et d’erreurs et ne nécessitent pas de configuration. Appuyez sur Ctrl+S pour sauvegarder votre Job.

**Configurer le composant tESBConsumer**

**Pourquoi et quand exécuter cette tâche**

Dans ce scénario, un Service Web public, disponible à l’adresse suivante http://www.webservicex.net/ValidateEmail.asmx est appelé par le composant tESBConsumer pour retourner true ou false pour une adresse e-mail. Vous pouvez voir la définition WSDL du Service à l’adresse http://www.webservicex.net/ValidateEmail.asmx?WSDL pour la description du Service.

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le tESBConsumer pour ouvrir sa vue Basic settings.
2. Cliquez sur le bouton [...] à côté de **Service configuration**.

3. Dans la boîte de dialogue qui s’ouvre, saisissez : `http://www.webservicex.net/ValidateEmail.asmx?WSDL` dans le champ **WSDL** puis cliquez sur le bouton de rafraîchissement afin de récupérer le nom du port et de l’opération. Dans la liste **Port Name**, sélectionnez le port que vous souhaitez utiliser, `ValidateEmailSoap` dans cet exemple.

Cochez la case **Populate schema to repository on finish** afin de récupérer la définition WSDL, qui sera utilisée par le composant tFixedFlowInput. Cette option est disponible pour les utilisateurs du Studio Talend avec ESB. Si vous n’avez pas cette option, ignorez la case. Le schéma peut être créé manuellement dans le composant tFixedFlowInput.
Cliquez sur Finish afin de valider vos paramètres et fermer la boîte de dialogue.


5. Cochez la case Log messages pour afficher le log de l'échange dans la console d'exécution.

**Configurer le tFixedFlowInput**

**Procédure**

1. Double-cliquez sur le tFixedFlowInput pour ouvrir sa vue Basic settings.


Les utilisateurs du Studio Talend sans ESB, vous pouvez passer à l'étape suivante.

Cliquez sur le bouton [...] à côté du champ Edit Schema. Dans la boîte de dialogue du schéma, cliquez sur le bouton [+] pour ajouter une ligne de type String et nommez-la Email. Cliquez sur OK pour fermer la boîte de dialogue.
4. Dans le champ **Number of rows**, configurez le nombre de lignes à 1.

5. Dans la zone **Mode**, sélectionnez **Use Single Table** et saisissez la requête suivante entre guillemets doubles dans le champ **Value** :

   nomatter@gmail.com

**Configurer le tXMLMap dans le flux d’entrée**

**Pourquoi et quand exécuter cette tâche**

L’intégration de données **Talend** utilise des schémas basés sur des lignes et colonnes puisque ses racines se trouvent dans l’intégration de data warehouses relationnels. Les messages SOAP utilisent le format XML. Le XML est hiérarchique et supporte des structures plus riches que les lignes ou les colonnes. Le **tXMLMap** doit donc convertir la structure relationnelle ligne/colonne en schéma attendu par le Service SOAP.

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le composant **tXMLMap** pour ouvrir son éditeur **Map Editor**.

2. Dans la table de sortie, cliquez-droit sur le nœud racine et sélectionnez **Rename** dans le menu contextuel. Saisissez *IsValidEmail* dans la boîte de dialogue qui apparaît.

3. Cliquez-droit sur le nœud *IsValidEmail* et sélectionnez **Set A Namespace** dans le menu contextuel. Saisissez **http://www.webservicex.net** dans la boîte de dialogue qui apparaît.

4. Cliquez-droit sur le nœud *IsValidEmail* à nouveau et sélectionnez **Create Sub-Element**. Saisissez *Email* dans la boîte de dialogue.

5. Cliquez-droit sur le nœud *Email* et sélectionnez **As loop element** dans le menu contextuel.

6. Cliquez sur le nœud *Email* de la table d’entrée et déposez-le dans la colonne **Expression**, dans la ligne du nœud *Email* dans la table de sortie.
7. Cliquez sur **OK** pour valider le mapping et fermer l’éditeur **Map Editor**.

**Configurer le composant tXMLMap dans le flux de sortie**

**Pourquoi et quand exécuter cette tâche**

Le **tXMLMap** du flux de sortie convertit le message de réponse du format XML en une structure ligne/colonne.

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le **tXMLMap** du flux de sortie pour ouvrir son éditeur **Map Editor**.
2. Dans la table d’entrée, cliquez sur le nœud racine et sélectionnez **Rename** dans le menu contextuel. Saisissez **IsValidEmailResponse** dans la boîte de dialogue qui s’ouvre.
3. Cliquez-droit sur le nœud **IsValidEmailResponse** et sélectionnez **Set A Namespace** dans le menu contextuel. Saisissez **http://www.webservicex.net** dans la boîte de dialogue qui s’ouvre.
4. Cliquez-droit sur le nœud **IsValidEmailResponse** et sélectionnez **Create Sub-Element** dans le menu contextuel. Enter **IsValidEmailResult** dans la boîte de dialogue qui s’ouvre.
5. Cliquez-droit sur le nœud **IsValidEmailResult** et sélectionnez **As loop element** dans le menu contextuel.
6. Dans la partie inférieure droite de l’éditeur de mapping, cliquez sur [*+] pour ajouter une ligne de type **String** à la table de sortie et nommez-la **response**.
7. Cliquez sur le nœud **IsValidEmailResult** dans la table d’entrée et déposez-le sur la colonne **Expression** dans la ligne du nœud **response**, dans la table de sortie.
8. Cliquez sur **OK** afin de valider le mapping et fermer le **Map Editor**.

**Exécuter le Job**

Cliquez sur l’onglet **Run** pour l’afficher et cliquez sur le bouton **Run** afin de lancer l’exécution du Job. Vous pouvez également appuyer sur **F6** pour l’exécuter. Dans la console, vous pouvez voir:
L'adresse e-mail nomatter@gmail.com est retournée comme false. Les messages SOAP d'entrée et de sortie en XML sont également affichés dans la console.

**Scénario 2 : Utiliser le tESBConsumer avec des en-têtes SOAP personnalisés**

Ce scénario est similaire au précédent. Le Job utilise un tESBConsumer pour récupérer une adresse e-mail valide ainsi que des en-têtes personnalisés SOAP dans le message de requête.
Déposer et relier les composants

Procédure
1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un tESBConsumer, un tMap, deux tFixedFlowInput, trois tXMLMap et deux tLogRow.
2. Reliez les composants tFixedFlowInput au tXMLMap à l’aide d’un lien Row > Main.
3. Cliquez-droit sur le premier tXMLMap, sélectionnez Row > *New Output* (Main) dans le menu contextuel et cliquez sur le tMap. Saisissez payload dans la boîte de dialogue qui s’ouvre, pour nommer cette connexion. Répétez l’opération pour relier un autre tXMLMap au tMap et nommez le lien de sortie header.
4. Cliquez-droit sur le composant tMap, sélectionnez Row > *New Output* (Main) dans le menu contextuel et cliquez sur le composant tESBConsumer. Saisissez request dans la boîte de dialogue qui s’ouvre, afin de nommer cette ligne et acceptez la propagation proposée afin d’obtenir le schéma du tESBConsumer.
5. Cliquez-droit sur le tESBConsumer, sélectionnez Row > Response dans le menu contextuel et cliquez sur le troisième tXMLMap.
6. Cliquez-droit sur le troisième tXMLMap, sélectionnez Row > *New Output* (Main) dans le menu contextuel et cliquez sur le premier tLogRow. Saisissez response dans la boîte de dialogue qui s’ouvre, afin de renommer cette connexion.
7. Cliquez-droit sur le tESBConsumer à nouveau, sélectionnez Row > Fault dans le menu contextuel puis cliquez sur l’autre tLogRow.

Configurer les composants

Les composants tLogRow monitorent les échanges des messages de réponse et d’erreur et ne nécessitent pas de configuration. Appuyez sur les touches Ctrl+S afin de sauvegarder votre Job.

Configurer le composant tESBConsumer

Pourquoi et quand exécuter cette tâche
Dans ce scénario, un Service Web public, disponible à l’adresse suivante http://www.webservicex.net/ValidateEmail.asmx est appelé par le composant tESBConsumer pour retourner true ou false pour une adresse e-mail. Vous pouvez voir la définition WSDL du Service à l’adresse http://www.webservicex.net/ValidateEmail.asmx?WSDL pour la description du Service.
Procédure

1. Dans l’espace de modélisation graphique, double-cliquez sur le composant **tESBConsumer** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton [...] à côté de **Service configuration**.

3. Dans la boîte de dialogue qui s’ouvre, saisissez : `http://www.webservicex.net/ValidateEmail.asmx?WSDL` dans le champ **WSDL** et cliquez sur le bouton de rafraîchissement, pour récupérer le nom du port et de l’opération. Dans la liste **Port Name**, sélectionnez le port à utiliser, **ValidateEmailSoap** dans cet exemple. Cliquez sur **OK** afin de valider les paramètres et de fermer la boîte de dialogue.

   Cochez la case **Populate schema to repository on finish** pour récupérer le schéma de la définition WSDL, qui sera utilisé par le composant **tFixedFlowInput**. Cette option est disponible uniquement pour les utilisateurs du **Studio Talend** avec ESB. Si vous n’avez pas l’option, ignorez la case. Le schéma peut être créé manuellement dans le composant **tFixedFlowInput**.
Cliquez sur Finish pour valider vos paramètres et fermer la boîte de dialogue.

4. Dans la vue Advanced settings, cochez la case Log messages pour enregistrer le contenu des messages.

Configurer les composants tFixedFlowInput

Procédure

1. Double-cliquez sur le premier tFixedFlowInput pour ouvrir sa vue Basic settings.

3. Pour les utilisateurs du *Studio Talend* sans ESB, le schéma doit être créé manuellement. Sélectionnez *Built-In* dans la liste *Schema*.

Cliquez sur le bouton [...] à côté du champ *Edit Schema*. Dans la boîte de dialogue du schéma, cliquez sur le bouton [+] pour ajouter une ligne de type *String* et nommez-la *Email*. Cliquez sur OK pour fermer la boîte de dialogue.
4. Dans le champ **Number of rows**, paramétrez le nombre de lignes à 1.

5. Dans la zone **Mode**, sélectionnez **Use Single Table** et saisissez "nomatter@gmail.com" dans le champ **Value**, le payload du message de requête.

6. Configurer le second **tFixedFlowInput** de la même façon, mais créez un schéma différent. Dans le schéma, ajoutez deux lignes de type **String** et nommez-les **id** et **company**, respectivement.

Donnez la valeur **Hello world!** à la colonne **id** et **Talend** à la colonne **company**, qui sont les en-têtes du message de requête.
Configurer les composants tXMLMap du flux d’entrée

Pourquoi et quand exécuter cette tâche

L’intégration de données Talend utilise des schémas basés sur des lignes et colonnes puisque ses racines se trouvent dans l’intégration de data warehouses relationnels. Les messages SOAP utilisent le format XML. Le XML est hiérarchique et supporte des structures plus riches que les lignes ou les colonnes. Le tXMLMap doit donc convertir la structure relationnelle ligne/colonne en schéma attendu par le Service SOAP.

Procédure

1. Dans l’espace de modélisation graphique, double-cliquez sur le premier tXMLMap pour ouvrir son éditeur Map Editor.
2. Dans la table de sortie, cliquez-droit sur le nœud racine et sélectionnez Rename dans le menu contextuel. Saisissez IsValidEmail dans la boîte de dialogue qui s’ouvre.
3. Cliquez-droit sur le nœud IsValidEmail et sélectionnez Set A Namespace dans le menu contextuel. Saisissez http://www.webservicex.net dans la boîte de dialogue qui s’ouvre.
4. Cliquez-droit à nouveau sur le nœud IsValidEmail et sélectionnez Create Sub-Element dans le menu contextuel. Saisissez Email dans la boîte de dialogue qui s’ouvre.
5. Cliquez-droit sur le nœud Email et sélectionnez As loop element dans le menu contextuel.
6. Cliquez sur le nœud Email dans la table d’entrée et déposez-le dans la colonne Expression de la ligne de nœud Email, dans la table de sortie.
7. Cliquez sur OK pour valider le mapping et fermer l’éditeur Map Editor.
Configurer le composant tMap

Procédure

1. Dans l’espace de modélisation graphique, double-cliquez sur le tMap pour ouvrir son Map Editor.

2. Dans la partie inférieure droite de l’éditeur, cliquez sur le bouton [+] pour ajouter deux lignes de type Document à la table de sortie et nommez-les respectivement payload et headers.

3. Cliquez sur le nœud payload dans la table d’entrée et déposez-le dans la colonne Expression de la ligne du nœud payload, dans la table de sortie.

4. Cliquez sur le nœud header dans la table d’entrée et déposez-le dans la colonne Expression de la ligne du nœud headers dans la table de sortie.

Configurer le tXMLMap du flux de sortie

Pourquoi et quand exécuter cette tâche

Le tXMLMap du flux de sortie convertit le message de réponse du format XML en une structure ligne/colonne.

Procédure

1. Dans l’espace de modélisation graphique, double-cliquez sur le tXMLMap du flux de sortie pour ouvrir son Map Editor.

2. Dans la table d’entrée, cliquez-droit sur le nœud et sélectionnez Rename dans le menu contextuel. Saisissez IsValidEmailResponse dans la boîte de dialogue qui s’ouvre.

4. Cliquez-droit sur le nœud IsValidEmailResponse à nouveau et sélectionnez **Create Sub-Element** dans le menu contextuel. Saisissez IsValidEmailResult dans la boîte de dialogue qui s’ouvre.

5. Cliquez-droit sur le nœud IsValidEmailResult et sélectionnez **As loop element** dans le menu contextuel.

6. Dans la partie inférieure droite de l’éditeur de mapping, cliquez sur le bouton [+] pour ajouter une ligne de type **String** à la table de sortie et nommez-la **response**.

7. Cliquez sur le nœud IsValidEmailResult dans la table d’entrée et déposez-le dans la colonne **Expression**, dans la ligne du nœud response de la table de sortie.

8. Cliquez sur **OK** pour valider le mapping et fermer l’éditeur **Map Editor**.

**Exécuter le Job**

Cliquez sur l’onglet **Run** pour afficher la vue du même nom. Cliquez sur le bouton **Run** pour lancer l’exécution de votre Job. Vous pouvez également appuyer sur **F6** pour l’exécuter.
Comme affiché dans le log d’exécution, l’adresse e-mail nomatter@gmail.com est retournée comme false. Les messages d’entrée et de sortie XML sont également affichés dans la console. L’en-tête SOAP est passé au service, dans la requête.
**tESBProviderFault**

Ce composant retourne un message Fault provenant du Service Web à la fin du cycle du Job *Talend* dans le cas d’une communication requête-réponse.

Le tESBProviderFault agit comme un message Fault (message d’erreur) de réponse du Service Web, à la fin du cycle d’un Job *Talend*.

**Propriétés du tESBProviderFault Standard**

Ces propriétés sont utilisées pour configurer le tESBProviderFault s’exécutant dans le framework de Jobs Standard.

Le composant tESBProviderFault Standard appartient à la famille ESB.

Ce composant est adapté pour une utilisation au sein de la perspective *Mediation* du Studio *Talend*. Il requiert l’utilisation du nœud du Repository *Service* et des assistants de création de *Services*.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur *Sync columns* afin de récupérer automatiquement les colonnes du composant précédent. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong> : Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du <em>Studio Talend</em>.</td>
<td></td>
</tr>
</tbody>
</table>
EBS service settings

Fault title : Valeur de la colonne faultString dans le message Fault.

Remarque :
Le flux Row > Fault du composant tESBConsumer contient un schéma prédéfini dont la colonne faultString contient les données du champ Fault title du tESBProviderFault.

Advanced settings

tStatCatcher Statistics

Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

Global Variables

Global Variables

NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant doit être utilisé uniquement avec le composant tESBProviderRequest.

Limitation

Une JDK est requise pour que ce composant puisse fonctionner.

Scénario : Effectuer une requête sur les nom d’aéroports à partir des codes pays

Ce scénario s’applique uniquement aux solutions Talend avec ESB.

Ce scénario contient deux Jobs, un fournisseur de service de données et un consommateur de service de données. Le premier Job écoute les requêtes du consommateur via le tESBProviderRequest, met en
correspondance le code pays contenu dans la requête avec une table d’une base de données MySQL contenant des paires code pays/aéroport via le tMap puis retourne le nom correct de l’aéroport via le tESBProviderResponse. Si aucune correspondance n’est trouvée, un message d’erreur est retourné via le tESBProviderFault. Le consommateur envoie des requêtes au fournisseur et reçoit les informations concernant les aéroports ou les erreurs via le tESBConsumer.

**Construire le fournisseur de service de données pour publier un service**

Le service de données airport a déjà été configuré sous le nœud Services de la vue Repository. Son schéma contient trois éléments majeurs :

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="airport"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.talend.org/service/"
targetNamespace="http://www.talend.org/service/">
  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.talend.org/service/">
      <xsd:element name="getAirportInformationbyISOCountryCodeRequest">
        <xsd:complexType>
          <xsd:sequence>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="getAirportInformationbyISOCountryCodeResponse">
        <xsd:complexType>
          <xsd:sequence>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="getAirportInformationbyISOCountryCodeFault">
        <xsd:complexType>
          <xsd:sequence>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
</wsdl:definitions>
```

Pour plus d’informations concernant la définition d’un service dans le studio, consultez le Guide utilisateur du Studio Talend.

**Assigner un Job au service défini**

**Procédure**

2. Dans la fenêtre [Operation Choice], sélectionnez Create a new Job and Assign it to this Service Operation.

![Assign Job](image)
3. Cliquez sur **Next** pour ouvrir la fenêtre de description de Job. Le nom du Job *airportSoap_ge_tAircraftInformationByISOCountryCode* est automatiquement renseigné.

4. Cliquez sur **Finish** pour créer le Job et l’ouvrir dans l’espace de modélisation graphique. Trois composants sont déjà disponibles.

**Ajouter des composants pour construire le flux de données**

**Procédure**

1. Déposez un **tXMLMap** et un **tMysqlInput** de la **Palette** dans l’espace de modélisation graphique.
2. Reliez le **tESBProviderRequest** au **tXMLMap** à l’aide d’un lien **Row > Main**.
3. Reliez le **tMysqlInput** au **tXMLMap** à l’aide d’un lien **Row > Main**.
4. Reliez le **tXMLMap** au **tESBProviderResponse** à l’aide d’un lien **Row > *New Output*(Main)**.
Dans la fenêtre [new Output name], saisissez le nom de la table de sortie, airport_response. Cliquez sur OK dans la fenêtre vous proposant de récupérer le schéma du composant cible.

5. Reliez le tXMLMap au tESBProviderFault à l’aide d’un lien Row > “New Output”(Main).
Dans la fenêtre [new Output name] qui s’ouvre, saisissez le nom de la table de sortie, fault_message.
Cliquez sur OK dans la fenêtre vous proposant de récupérer le schéma du composant suivant.

Configurer la manière dont sont traitées les requêtes

Procédure

1. Double-cliquez sur le tMysqlInput pour afficher sa vue Basic settings.

2. Renseignez les propriétés de base de la connexion MySQL et de la table de base de données. Cliquez sur le bouton [...] pour ouvrir l’éditeur de schéma.
3. Cliquez deux fois sur le bouton [+] pour ajouter deux colonnes de type String. Nommez-les respectivement *id* et *name*.
   Cliquez sur **OK** pour fermer l’éditeur.
   Cliquez sur **Guess Query** pour récupérer la requête SQL.

4. Double-cliquez sur le **tXMLMap** pour ouvrir l’éditeur de mapping.

5. Dans la table **main : row1** du côté de l’entrée (gauche), cliquez-droit sur le nom de la colonne *payload* et, dans le menu contextuel, sélectionnez **Import from Repository**. L’assistant [Metadata] s’ouvre.
Sélectionnez le schéma du message de requête et cliquez sur OK pour valider la sélection. Dans cet exemple, le schéma est `getAirportInformationByISOCountryCode`.


7. Pour créer la jointure sur les données de référence, déposez le nœud `CountryAbbrv` de la table principale sur la ligne `id` de la table de référence (lookup).

8. Dans la table de référence, cliquez sur l'icône de clé anglaise, dans le coin supérieur droit, pour ouvrir le panneau des paramètres.

   Paramétrez **Lookup Model** à **Reload at each row**, **Match Model** à **All matches** et **Join Model** à **Inner join**.

9. Dans la table du flux de sortie `airport_response`, cliquez sur l'icône de clé anglaise, dans le coin supérieur droit, pour ouvrir le panneau des paramètres.

   Paramétrez l'option **All in one** à **true**. Cela vous assure qu'une seule réponse est retournée pour chaque requête si plusieurs aéroports correspondent dans la base de données.

10. Dans la table de sortie `fault_message`, cliquez sur l'icône de clé anglaise, dans le coin supérieur droit, pour ouvrir le panneau des paramètres.

    Paramétrez l'option **Catch Lookup Inner Join Reject** à **true** pour montrerer les non-correspondances entre le code pays de la requête et les enregistrements dans la table de base de données. Lorsqu'une telle situation se produit, un message Fault est généré par le `tESBConsumer` et écrit en sortie par le flux `Row > Fault`.
Remarque :
Le flux Row > Fault du tESBConsumer contient un schéma prédéfini dont la colonne faultString contient les données du champ Fault title du tESBProviderFault.


Cliquez sur OK pour fermer l’éditeur et valider la configuration.

12. Double-cliquez sur le tESBProviderFault pour afficher sa vue Basic settings.


Pour plus d’informations concernant la définition des variables de contexte, consultez le Guide utilisateur du Studio Talend.

Publier le Service pour écouter les requêtes

Procédure

1. Appuyez sur les touches Ctrl +S pour sauvegarder le Job.

2. Appuyez sur F6 pour exécuter le Job.

Résultats
Le service de données est publié et écoute toutes les requêtes jusqu’à ce que vous cliquez sur le bouton Kill pour l’arrêter, puisque, par défaut, l’option Keep listening du tESBProviderRequest est automatiquement sélectionnée.

Configurez le Job consommateur qui interagit avec le service de données.
Construire le consommateur de service de données pour envoyer une requête au service

Construit autour du tESBConsumer, le Job consommateur envoie deux requêtes contenant des codes pays au service Web pour obtenir le nom des aéroports correspondants. Si un mauvais code pays est contenu dans la requête, un message d’erreur est retourné. Les codes pays et les enregistrements de la base de données MySQL se présentent comme suit :

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN</td>
<td>Capital Airport</td>
</tr>
<tr>
<td>DE</td>
<td>Frankfurt Airport</td>
</tr>
<tr>
<td>FR</td>
<td>Roissy Airport</td>
</tr>
<tr>
<td>(NULL)</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

Déposer et relier les composants

**Procédure**

1. Déposez un tFileInputDelimited, un tXMLMap, un tESBConsumer et deux tLogRow de la Palette dans l’espace de modélisation graphique.
3. Reliez le tFileInputDelimited au tXMLMap à l’aide d’un lien Row > Main.
4. Reliez le tXMLMap au tESBConsumer à l’aide d’un lien Row > "New Output"(Main).
   Dans la fenêtre [new Output name] qui s’ouvre, saisissez le nom de la table de sortie, par exemple request.
   Cliquez sur OK dans la fenêtre vous proposant de récupérer le schéma du composant cible.
5. Reliez le tESBConsumer au composant response à l’aide d’un lien Row > Response.
6. Reliez le tESBConsumer au fault_message à l’aide d’un lien Row > Fault.

Configurer les composants

**Procédure**

1. Double-cliquez sur le tFileInputDelimited pour ouvrir sa vue Basic settings.
2. Dans le champ **File name/stream**, saisissez la variable de contexte pour le fichier contenant les codes pays, par exemple `context/filepath`.

3. Cliquez sur le bouton `[...]` pour ouvrir l'éditeur de schéma.

4. Cliquez sur le bouton `[+]` pour ajouter une colonne `country_code`, de type String. Cliquez sur **OK** pour fermer l'éditeur.

5. Double-cliquez sur le **tXMLMap** pour ouvrir son éditeur de mapping.

 Sélectionnez le schéma du message de requête et cliquez sur OK pour valider la sélection. Dans cet exemple, le schéma est getAirportInformationByISOCountryCode.
7. Déposez la colonne *country_code* du flux principal dans la zone *Expression*, à côté du nœud *tns:CountryAbbreviation* dans le flux de sortie *request*. Cliquez sur OK pour fermer l’éditeur et valider la configuration.

8. Double-cliquez sur le composant *tESBConsumer* pour ouvrir son assistant de configuration de service.

9. Cliquez sur le bouton *Browse...* pour sélectionner le fichier WSDL défini. Les champs *Port name* et *Operation* sont automatiquement renseignés, une fois le fichier WSDL sélectionné. Cliquez sur OK pour fermer l’assistant.

10. Double-cliquez sur le composant *response* pour ouvrir sa vue *Basic settings*.

11. Sélectionnez l’option *Vertical (each row is a key/value list)* puis *Print label* afin de visualiser les résultats de manière optimale. Répétpez l’opération pour l’autre *tLogRow, fault_message*.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl + S pour sauvegarder le Job.
2. Appuyez sur F6 pour exécuter ce Job.
Comme affiché ci-dessus, deux messages sont retournés, le premier donnant le nom de l’aéroport correspondant au code pays CN et le second donnant les détails de l’erreur causée par le code pays CC.
**tESBProviderRequest**

Expose un Job **Talend** comme un Service Web.

Le tESBProviderRequest attend un message de requête d'un consommateur et le passe au composant suivant.

**Propriétés du tESBProviderRequest Standard**

Ces propriétés sont utilisées pour configurer le tESBProviderRequest s'exécutant dans le framework de Jobs Standard.

Le composant tESBProviderRequest Standard appartient à la famille ESB.

Ce composant est adapté pour une utilisation au sein de la perspective **Mediation** du Studio Talend. Il requiert l'utilisation du nœud du Repository **Service** et des assistants de création de **Services**.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Aucun fichier WSDL n'est configuré pour le Job.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de service Web souhaité depuis le Repository pour sélectionner les paramètres (granularité) du nom du port et de l'opération.</td>
</tr>
</tbody>
</table>

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.


**Keep listening**

Cochez cette case lorsque vous souhaitez assurer que le fournisseur (le Job *Talend*) continue à écouter les requêtes après traitement de la première requête entrante.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Log messages (Studio only)</strong></th>
<th>Cochez cette case pour enregistrer l’échange de messages entre le service et le consommateur. Cette option fonctionne uniquement avec le Studio.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Response timeout, sec</strong></td>
<td>Spécifiez la limite de temps en secondes pour envoyer une réponse au consommateur. Ce paramètre est nécessaire pour éviter le verrouillage des échanges de messages.</td>
</tr>
<tr>
<td><strong>Request processing queue size</strong></td>
<td>Spécifiez le nombre maximum de requêtes reçues qui peuvent être traitées en parallèle par les composants entre le <em>tESBProviderRequest</em> et le <em>tESBProviderResponse</em>. Ce paramètre est différent du <em>queueSize</em> dans <code>&lt;TalendRuntimePath&gt;/etc/org.apache.cxf.workqueues-default.cfg</code>, qui définit la configuration du pool pour les requêtes entrantes au niveau de CXF.</td>
</tr>
<tr>
<td><strong>Request processing timeout, sec</strong></td>
<td>Spécifiez la limite de temps en secondes pour le traitement des requêtes par les composants entre le <em>tESBProviderRequest</em> et le <em>tESBProviderResponse</em>.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th><strong>Global Variables</strong></th>
<th><strong>NB_LINE</strong> : nombre de lignes traitées. Cette variable est une variable <em>After</em> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>CORRELATION_ID</strong> : l’ID de corrélation par lequel re糖尿er les appels de services en chaîne. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>SECURITY_TOKEN</strong> : informations concernant l’identité de l’utilisateur dans l’en-tête de requête. Cette variable est une variable <em>Flow</em> et retourne un nœud XML.</td>
</tr>
<tr>
<td></td>
<td><strong>HEADERS_SOAP</strong> : en-têtes de la requête SOAP. Cette variable est une variable <em>Flow</em> et retourne tous les en-têtes des requêtes SOAP.</td>
</tr>
<tr>
<td></td>
<td><strong>HEADERS_HTTP</strong> : en-têtes de la requête HTTP. Cette variable est une variable <em>Flow</em> et retourne tous les en-têtes des requêtes HTTP.</td>
</tr>
</tbody>
</table>
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

| Règle d’utilisation | Ce composant couvre la possibilité d’exposer un Job **Talend** en tant que Service, avec la possibilité d’écrire une requête de service dans un Job et retourner le résultat du Job en tant que réponse. Le composant **tESBProviderResponse** peut livrer le payload d’un message SOAP et accéder aux en-têtes HTTP et SOAP d’un service. Le composant **tESBProviderRequest** doit être utilisé avec le **tESBProviderResponse** afin de fournir une réponse à la fin du Job, dans le cas d’une communication de style requête-réponse. Lorsque SAML Token ou Service Registry est activé dans les options du Runtime et si le jeton SAML existe dans l’en-tête de la requête, le **tESBProviderRequest** obtient et stocke le jeton SAML dans la variable du composant pour un usage ultérieur dans le flux. Le **tESBProviderRequest** obtient une valeur de corrélation (Correlation Value) dans l’en-tête de la requête si la valeur existe et qu’elle est stockée dans la variable du composant. Lorsqu’une corrélation métier (Business Correlation) ou le registre de Service (Service Registry) est activé(e) dans les options du Runtime, la **Correlation Value** est également ajoutée à la réponse. Dans ce cas, le **tESBProviderRequest** crée une Correlation Value si elle n’existe pas. Notez que l’option de Service Registry est disponible uniquement si vous avez souscrit à l’une des solutions Enterprise ESB de **Talend**. Pour plus d’informations concernant les options du Runtime, consultez la section correspondante dans le **Guide utilisateur du Studio Talend**. |
| Dynamic settings | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin d’activer ou désactiver dynamiquement l’option **Keep listening** au moment de l’exécution. |

**tESBProviderRequest**
| Limitation | Une JDK est requise pour que ce composant fonctionne. |

**Scénario : Envoi d’un message via un service sans attente de réponse**

Ce scénario s’applique uniquement aux solutions Talend avec ESB.

Les Jobs suivants, construits sur les composants de la famille **ESB/Web Service** implémentent des services Web définis dans le nœud **Services** du Repository. Ils requièrent la création et l’association des services correspondants. Pour plus d’informations concernant les services, consultez le **Guide utilisateur du Studio Talend**.

Dans ce scénario, un Job fournisseur et un Job consommateur sont nécessaires. Par ailleurs, le service correspondant doit déjà exister sous le nœud **Services**, avec l’URI `http://127.0.0.1:8088/esb/provider/?WSDL`, le port `TEST_ProviderJobSoapBinding` et l’opération `invoke(anyType):anyType`.

Le Job fournisseur comprend les composants suivants : un **tESBProviderRequest**, un **tXMLMap** et deux **tLogRow**.

- Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un **tESBProviderRequest**, un **tXMLMap** et deux **tLogRow**.
- Double-cliquez sur le composant **tESBProvider Request** afin d’ouvrir sa vue **Basic settings**.
• Sélectionnez **Repository** dans la liste **Property Type** et cliquez sur le bouton […] afin de choisir le service, la granularité du port et l’opération.

• Cliquez sur **OK**.

• Cliquez sur le bouton […] à côté du champ **Edit schema** pour voir le schéma du composant **tESBProvider Request**.

• Cliquez sur **OK**.

• Connectez le **tESBProviderRequest** au composant **tLogRow_1**.

• Double-cliquez sur le **tLogRow_1** dans l’espace de modélisation graphique afin d’afficher sa vue **Basic settings**.
• Cliquez sur le bouton [...] à côté du champ Edit schema et configurez le schéma comme suit.

• Connectez le tLogRow_1 au tXMLMap.

• Reliez le tXMLMap et tLogRow_2 et nommez la connexion payload.

• Dans l’espace de modélisation graphique, double-cliquez sur le tXMLMap afin d’ouvrir le Map Editor.

• Dans la partie inférieure droite de l’éditeur, cliquez sur le bouton [+] pour ajouter une ligne à la table payload et nommez cette ligne payload.

• Dans la colonne Type de cette ligne payload, sélectionnez le type de données Document. La racine XML correspondante est ajoutée automatiquement à la table supérieure de droite, représentant le flux de sortie.

• Dans la table payload, cliquez-droit sur root pour ouvrir son menu contextuel.

• Dans le menu contextuel, sélectionnez Create Sub-Element et saisissez response dans la boîte de dialogue.

• Cliquez-droit sur le nœud response et sélectionnez As loop element dans le menu contextuel.

• Répétez l’opération pour créer un sous-élément request du noeud root dans la table d’entrée et définissez le nœud request comme élément de boucle.

• Cliquez sur le nœud request dans la table d’entrée et déposez-le dans la colonne Expression de la ligne du nœud response de la table de sortie.
• Cliquez sur OK pour valider le mapping et fermer l'éditeur.

• Double-cliquez sur le **tLogRow_2** dans l'espace de modélisation graphique afin d'afficher sa vue **Basic settings**.

• Cliquez sur le bouton [...] à côté du champ **Edit schema** et configurez le schéma comme suit.
• Sauvegardez le Job.

Le Job consommateur comprend un **tFixedFlowInput**, un **tXMLMap**, un **tESBConsumer** et deux **tLogRow**.

• Déposez les composants suivants de la **Palette** dans l’espace de modélisation graphique : un **tFixedFlowInput**, un **tXMLMap**, un **tESBConsumer** et deux **tLogRow**.

• Double-cliquez sur le **tFixedFlowInput** afin d’afficher sa vue **Basic settings**.
• Configurez le schéma du composant **tFixedFlowInput**.

![Schema de tFixedFlowInput](image)

• Cliquez sur le bouton `[+]` pour ajouter une ligne de type **String** et nommez-la *payloadString*.

• Cliquez sur **OK**.

• Dans le champ **Number of rows**, configurez le nombre de lignes à 1.

• Dans la zone **Mode**, sélectionnez **Use Single Table** et saisissez *world* entre guillemets, dans le champ **Value**.

• Connectez le **tFixedFlowInput** au **tXMLMap**.

• Reliez le **tXMLMap** au **tESBConsumer** et nommez la connexion *payload*.

• Dans l’espace de modélisation graphique, double-cliquez sur le **tXMLMap** pour ouvrir le **Map Editor**.

• Dans la table de sortie, cliquez-droit sur *root* pour ouvrir son menu contextuel.

• Dans le menu contextuel, sélectionnez **Create Sub-Element** et saisissez *request* dans la boîte de dialogue.

• Cliquez-droit sur le nœud *request* et sélectionnez **As loop element** dans le menu contextuel.

• Cliquez sur le nœud *payloadstring* dans la table d’entrée et déposez-le dans la colonne **Expression** dans la ligne du nœud *request* de la table de sortie.
• Cliquez sur **OK** pour valider le mapping et fermer l'éditeur.

• Démarrrez le Job fournisseur. Dans le log d'exécution, vous pouvez voir :

```plaintext
... web service [endpoint: http://127.0.0.1:8088/esb/provider] published ...
```

• Dans la vue **Component** du **tESBConsumer**, configurez ses propriétés.

• Cliquez sur le bouton [...] à côté du champ **Service Configuration** afin d'ouvrir l'éditeur.
Dans le champ **WSDL**, saisissez : http://127.0.0.1:8088/esb/provider?WSDL

Cliquez sur le bouton **Refresh** afin de récupérer le nom du port et de l’opération.

Cliquez sur **OK**.

Configurez le schéma d’entrée (**Input Schema**) du composant **tESBConsumer** comme suit.

Configurez le schéma de réponse (**Response Schema**) comme suit.

Configurez le schéma **Fault** comme suit.

Connectez le composant **tESBConsumer** au **tLogRow_1** et au **tLogRow_2**.
• Dans l’espace de modélisation graphique, double-cliquez sur le `tLogRow_1` afin d’afficher sa vue `Basic settings`.

![Diagramme de tLogRow_1](image1)

• Cliquez sur le bouton à côté du champ `Edit schema` et configurez le schéma comme suit.

![Schéma de tLogRow_1](image2)

• Dans l’espace de modélisation graphique, double-cliquez sur `tLogRow_2` afin d’afficher sa vue `Basic settings`.

![Diagramme de tLogRow_2](image3)

• Cliquez sur le bouton ` [...] ` à côté du champ `Edit schema` et configurez le schéma comme suit.

![Schéma de tLogRow_2](image4)
• Sauvegardez le Job.

• Exécutez le Job fournisseur. Vous pouvez voir dans le log d’exécution :

INFO: Setting the server’s publish address to be http://127.0.0.1:8088/esb/provider
2011-04-21 14:14:36.793:INFO::jetty-7.2.2.v20101205
2011-04-21 14:14:37.856:INFO::Started
SelectChannelConnector@127.0.0.1:8088
web service [endpoint: http://127.0.0.1:8088/esb/provider] published

• Exécutez le Job consommateur. Vous pouvez voir dans le log d’exécution du Job :

```java
Starting job CallProvider at 14:15 21/04/2011.
[statistics] connecting to socket on port 3942
[statistics] connected
TEST_ESBProvider2
TEST_ESBProvider2SoapBinding
 | [tLogRow_2] payloadString: <request>world</request>
(http://talend.org/esb/service/job)TEST_ESBProvider2
(http://talend.org/esb/service/job)TEST_ESBProvider2SoapBinding
invoke
[tLogRow_1] payload: null
[statistics] disconnected
Job CallProvider2 ended at 14:16 21/04/2011. [exit code=0]
```

• Dans le log du fournisseur vous pouvez voir le log des traces :

```java
web service [endpoint: http://127.0.0.1:8088/esb/provider] published
[tLogRow_1] payload: <?xml version="1.0" encoding="UTF-8"?>
<request>world</request>
### world
[tLogRow_2] content: world
[tLogRow_3] payload: <?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://talend.org/esb/service/job">Hello, world!</response>
[statistics] disconnected
Job ESBProvider2 ended at 14:16 21/04/2011. [exit code=0]
```
**tESBProviderResponse**

Ce composant retourne un message de réponse à la fin de chaque cycle d’un Job **Talend**.

Le tESBProviderResponse génère un message de réponse à la fin de chaque cycle d’un Job **Talend**.

**Propriétés du tESBProviderResponse Standard**

Ces propriétés sont utilisées pour configurer le tESBProviderResponse s'exécutant dans le framework de Jobs Standard.

Le composant tESBProviderResponse Standard appartient à la famille ESB.

Ce composant est adapté pour une utilisation au sein de la perspective **Mediation** du **Studio Talend**. Il requiert l'utilisation du nœud du Repository **Service** et des assistants de création de **Services**.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

  - **View schema** : sélectionnez cette option afin de voir le schéma.
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur Sync columns afin de récupérer automatiquement les colonnes du composant précédent. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
## Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

## Global Variables

### NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

### ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

## Utilisation

### Règle d’utilisation

Le composant `tESBProviderRequest` doit être utilisé avec le `tESBProviderRequest` afin de fournir une réponse à la fin d’un Job, pour un fournisseur de service, dans le cas d’une communication de style requête-réponse.

### Limitation

Une JDK est requise pour que ce composant fonctionne.

## Scénario : Retourner une réponse "Hello world"

Ce scénario s’applique uniquement aux solutions Talend avec ESB.


Dans ce scénario, un Job fournisseur et un Job consommateur sont nécessaires. Par ailleurs, le service correspondant doit déjà exister sous le nœud Services, avec l’URI `http://127.0.0.1:8088/esb/provider/?WSDL`, le port `TEST_ProviderJobSoapBinding` et l’opération `invoke(anyType):anyType`.

Le Job fournisseur comprend les composants suivants : un `tESBProviderRequest`, un `tESBProviderResponse`, un `tXMLMap` et deux `tLogRow`. 
- Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : un tESBProviderRequest, un tESBProviderResponse, un XMLMap et deux tLogRow.
- Dans l'espace de modélisation graphique, double-cliquez sur le composant tESBProviderRequest afin d'afficher sa vue Basic settings.

- Dans la liste Property Type, sélectionnez Repository et cliquez sur le bouton [...] afin de sélectionner le service, la granularité du port et l'opération.

- Cliquez sur OK.
- Cliquez sur le bouton [...] à côté du champ Edit schema afin de voir le schéma.
• Connectez le **tESBProviderRequest** au **tLogRow_1**.

• Double-cliquez sur le **tLogRow_1** afin d'afficher sa vue **Basic settings**.

• Cliquez sur le bouton [...] à côté du champ **Edit schema** et configurez le schéma comme suit.

• Reliez le **tLogRow_1** au **tXMLMap**.

• Reliez le **tXMLMap** au **tLogRow_2** et nommez la connexion **payload**.

• Dans l'espace de modélisation graphique, double-cliquez sur le **tXMLMap** afin d'ouvrir le **Map Editor**.
• Dans la partie inférieure droite de l’éditeur, cliquez sur le bouton [+ ] pour ajouter une ligne à la table payload. Nommez cette ligne payload.

• Dans la colonne Type de la ligne payload, sélectionnez le type de données Document. La racine XML correspondante est ajoutée automatiquement à la table supérieure de droite, représentant le flux de sortie.

• Dans la table payload, cliquez-droit sur root pour ouvrir le même menu contextuel.

• Dans le menu contextuel, sélectionnez Create Sub-Element et saisissez response dans la boîte de dialogue.

• Cliquez-droit sur le nœud response et sélectionnez As loop element dans le menu contextuel.

• Répétez l’opération pour créer un sous-élément request du nœud root dans la table d’entrée et définissez le nœud request comme élément de boucle.

• Cliquez sur OK pour valider le mapping et fermer l’éditeur

• Dans l’espace de modélisation graphique, double-cliquez sur le tLogRow_2 afin d’afficher sa vue Basic settings.
- Cliquez sur le bouton [...] à côté du champ **Edit schema** et configurez le schéma comme suit.

- Reliez le **tLogRow_2** au **tESBProviderResponse**.

- Dans l’espace de modélisation graphique, double-cliquez sur le **tESBProviderResponse** pour ouvrir sa vue **Component** et configurer ses **Basic settings**.

- Cliquez sur le bouton [+] à côté du champ **Edit schema** et définissez le schéma comme suit.

- Sauvegardez le Job fournisseur.
Le Job consommateur comprend un tFixedFlowInput, un tXMLMap, un tESBConsumer et deux tLogRow.

- Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un tFixedFlowInput, un tXMLMap, un tESBConsumer et deux tLogRow.

- Double-cliquez sur le composant tFixedFlowInput afin d’afficher sa vue Basic settings.

- Cliquez sur le bouton […] à côté du champ Edit schema afin d’éditer le schéma.
• Cliquez sur le bouton [+ ] pour ajouter une nouvelle ligne de type String et nommez-la payloadString.

• Cliquez sur OK.

• Dans le champ Number of rows, définissez le nombre de lignes à 1.

• Dans la zone Mode, sélectionnez Use Single Table et saisissez world entre guillemets, dans le champ Value.

• Reliez le tFixedFlowInput au tXMLMap.

• Connectez le tXMLMap au tESBConsumer et nommez la connexion payload.

• Dans l’espace de modélisation graphique, double-cliquez sur le tXMLMap pour ouvrir le Map Editor.

• Dans la table de sortie, cliquez-droit sur root pour ouvrir son menu contextuel.

• Dans le menu contextuel, sélectionnez Create Sub-Element et saisissez request dans la boîte de dialogue.

• Cliquez-droit sur le nœud request et sélectionnez As loop element dans le menu contextuel.

• Cliquez sur le nœud payloadstring de la table d’entrée et déposez-le dans la colonne Expression de la ligne du nœud request dans la table de sortie.

• Cliquez sur OK pour valider le mapping et fermer le Map Editor.

• Démarrez le Job fournisseur. Vous pouvez voir, dans le log d’exécution :

```plaintext
... web service [endpoint: http://127.0.0.1:8088/esb/.provider] published ...
```

• Dans la vue Component du tESBConsumer, configurez les propriétés du composant.
Cliquez sur le bouton [...] à côté du champ **Service Configuration** pour ouvrir l’éditeur.

Dans le champ **WSDL**, saisissez : http://127.0.0.1:8088/esb/provider?WSDL

Cliquez sur le bouton **Refresh** afin de récupérer le nom du port et de l’opération.

Cliquez sur **OK**.

Dans la vue **Basic settings** du composant **tESBConsumer**, configurez le schéma d’entrée (Input Schema) comme suit.
• Configurez le schéma de réponse (*Response Schema*) comme suit.

• Configurez le schéma (*Fault Schema*) comme suit.

• Connectez le *tESBConsumer* aux deux composants *tLogRow*.

• Dans l’espace de modélisation graphique, double-cliquez sur le composant *tLogRow_1* afin d’ouvrir sa vue *Basic settings*.

• Cliquez sur le bouton [...] à côté du champ *Edit schema* et configurez le schéma comme suit.
• Dans l’espace de modélisation graphique, double-cliquez sur le tLogRow_2 afin d’afficher sa vue Basic settings.

![Diagram of tLogRow_2 settings]

• Cliquez sur le bouton [...] à côté du champ Edit schema et configurez le schéma comme suit.

![Diagram of tLogRow_2 output settings]

• Sauvegardez le Job consommateur.

• Exécutez le Job fournisseur. Vous pouvez voir, dans le log d’exécution :

```
2011-04-21 15:28:26.874:INFO::jetty-7.2.2.v20101205
SelectChannelConnector@127.0.0.1:8088
web service [endpoint: http://127.0.0.1:8088/esb/provider] published
```

• Exécutez le Job consommateur. Vous pouvez voir, dans le log d’exécution du Job :

```
Starting job CallProvider at 14:15 21/04/2011.
[statistics] connecting to socket on port 3942
[statistics] connected
TEST_ESBProvider2
TEST_ESBProvider2SoapBingding
tLogRow_2 payloadString: <request>world</request>{http://talend.org/esb/service/job}TEST_ESBProvider2
{http://talend.org/esb/service/job}TEST_ESBProvider2SoapBinding
invoke
tLogRow_1 payload: null
[statistics] disconnected
Job CallProvider2 ended at 14:16 21/04/2011. [exit code=0]
```

• Dans le log fournisseur, vous pouvez voir les traces :

```
web service [endpoint: http://127.0.0.1:8088/esb/provider]
```
published
[tLogRow_1] payload: <?xml version="1.0" encoding="UTF-8"?>
<request>world</request>
### world
[tLogRow_2] content: world
[tLogRow_3] payload: <?xml version="1.0" encoding="UTF-8"?>
<response xmlns="http://talend.org/esb/service/job">Hello, world!</response>
[statistics] disconnected
Job ProviderJob ended at 15:29 21/04/2011. [exit code=0]
tEXABulkExec

Ce composant importe rapidement des données dans une table de base de données EXASolution, à l’aide de la commande IMPORT fournie par la base de données EXASolution.

L’import est annulé lorsque l’import échoue pour un certain nombre d’enregistrements. Ce nombre est configurable. Les enregistrements en erreur peuvent être envoyés dans une table de log dans la même base de données ou dans un fichier de log local.

Propriétés du tEXABulkExec Standard

Ces propriétés sont utilisées pour configurer le tEXABulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tEXABulkExec Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et, dans la liste qui s’affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue <strong>Basic settings</strong> du composant de connexion créant cette connexion.</td>
</tr>
<tr>
<td></td>
<td>2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être <strong>Built-In</strong> ou <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

| Host | Saisissez l’hôte ou la liste d’hôtes des serveurs de bases de données EXASol. EXASol peut s’exécuter dans un environnememt de cluster. Une valeur valide peut être |
une adresse IP simple (par exemple 172.16.173.128), une liste d'adresses IP (par exemple 172.16.173.128..130 qui représente trois serveurs 172.16.173.128, 172.16.173.129, et 172.16.173.130), ou une liste d'hôtes séparés par des virgules (par exemple server1,server2,server3) du cluster de base de données EXASolution.

<table>
<thead>
<tr>
<th>Port</th>
<th>Saisissez le numéro du port d'écoute du cluster de base de données EXASolution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Saisissez le nom du schéma que vous souhaitez utiliser.</td>
</tr>
<tr>
<td>User et Password</td>
<td>Saisissez les données d'authentification de l'utilisateur pour accéder à la base de données EXASolution. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| Table | Nom de la table à créer.  
Remarque :  
Généralement, les noms de tables sont stockés en majuscules. Si vous utilisez des identifiants contenant des minuscules et des majuscules, saisissez le nom entre guillemets doubles. Par exemple, "\"TEST_data_LOAD\"". |
| Action on table | Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée avant import :  
• None : n'effectuer aucune opération de table.  
• Drop and create table : supprimer la table puis en créer une nouvelle.  
• Create table : créer une table qui n'existe pas encore.  
• Create table if not exists : créer la table si nécessaire.  
• Truncate table : supprimer le contenu de la table, sans possibilité de rollback. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
• Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
• Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.  
Remarque :  
Les colonnes du schéma doivent être dans le même ordre que dans le fichier .csv. Il n'est pas nécessaire de renseigner toutes les colonnes. |
de la table définie, sauf si le scénario ou la définition de la table attend ce renseignement.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Advanced settings

#### Additional JDBC Parameters

Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, `encryption=1;clientname=Talend`.

Ce champ n’est pas disponible si la case **Use an existing connection** est cochée.

#### Column Formats

Spécifiez le format pour les colonnes de type **Date** et **Numeric** si le type par défaut ne peut être appliqué.

- **Column** : les cellules de cette colonne sont automatiquement renseignées par les noms définis des colonnes du schéma.
- **Has Thousand Delimiters** : cochez cette case si la valeur de la colonne **Numeric** correspondante (colonnes **Numeric** uniquement) du fichier contient des séparateurs de milliers.
- **Alternative Format** : spécifiez le format nécessaire comme une valeur **String** si un format spécial est attendu. Le format nécessaire sera créé à partir de la longueur de la précision de la colonne du schéma. Pour plus d’informations concernant les modèles de format, consultez **EXASolution User Manual** (en anglais).

#### Source table columns

Si la source est une base de données, configurez dans cette table le mapping entre les colonnes source et cible.

La configuration du mapping est facultative. Si vous ne configurez rien ici, il est supposé que la table source a la même structure que la base de données cible.
| **tEXABulkExec** |
|-------------------|-----------------|
| **Column** : colonne du schéma dans la table cible. |
| **Source column name** : nom de la colonne dans la table source. |
| **Column Separator** : Saisissez le séparateur des colonnes au sein d’une ligne dans le fichier local. |
| **Column Delimiter** : Saisissez le séparateur de contenu des champs dans le fichier local. |
| **Row Separator** : Saisissez le caractère utilisé pour séparer les lignes dans le fichier local. |
| **Null representation** : Saisissez la chaîne de caractères représentant une valeur NULL dans le fichier local. Si rien n’est spécifié, les valeurs nulles sont représentées par une chaîne de caractères vide. |
| **Skip rows** : Saisissez le nombre de lignes à omettre (par exemples un en-tête ou d’autres préfixes). |
| **Encoding** : Saisissez le jeu de caractères utilisé dans le fichier local. Par défaut, l’encodage est **UTF8**. |
| **Trim column values** : Spécifiez si les espaces doivent être coupés au bord des colonnes .csv. |
| • **No trim** : ne couper aucun espace. |
| • **Trim** : couper les espaces à gauche et à droite. |
| • **Trim only left** : ne couper que les espaces à gauche. |
| • **Trim only right** : ne couper que les espaces à droite. |
| **Default Date Format** : Spécifiez le format de la date. Par défaut, le format est **YYYY-MM-DD**. |
| **Default Timestamp Format** : Spécifiez le format de l’heure système. Par défaut, le format est **YYYY-MM-DD HH24:MI:SS.FF3**. |
| **Thousands Separator** : Spécifiez le caractère utilisé pour séparer les groupes de milliers dans une valeur numérique. Au format **Numeric**, le caractère sera appliqué sur la valeur fictive G. Si le tes valeurs de texte contiennent ce caractère, vous devez également le configurer dans la table **Column Formats**. |
| Notez que ce paramètre affecte la propriété de connexion **NLS_NUMERIC_CHARACTERS** définissant les caractères décimaux et les caractères de groupe utilisés pour représenter les nombres. |
| **Decimal Separator** : Spécifiez le caractère utilisé pour séparer les parties entières des nombres des parties décimales. Au format **Numeric**, le caractère sera appliqué sur la valeur fictive D. |
| Notez que ce paramètre affecte la propriété de connexion **NLS_NUMERIC_CHARACTERS** définissant les caractères décimaux et les caractères de groupe utilisés pour représenter les nombres. |
### Minimal number errors to reject the transfer

Spécifiez le nombre maximum de lignes non valides autorisées lors du traitement de chargement des données. Par exemple, la valeur 2 signifie que le processus de chargement s’arrête si une troisième erreur survient.

### Log Error Destination

Spécifiez l’emplacement où vous souhaitez conserver les messages d’erreur.

- **No Logging**: les messages d’erreur ne sont pas sauvegardés.
- **Local Log File**: les messages d’erreur sont stockés dans un fichier local spécifié.
  - **Local Error Log File**: spécifiez le chemin d’accès au fichier local stockant les messages d’erreur.
  - **Add current timestamp to log file name (before extension)**: cochez cette case pour ajouter la date et l’heure courantes avant l’extension du fichier pour des raisons d’identification, si vous utilisez plusieurs fois le même fichier.
- **Logging Table**: les messages d’erreur sont sauvegardés dans une table spécifiée. Cette table est créée si elle n’existe pas.
  - **Error Log Table**: saisissez le nom de la table stockant les messages d’erreur.
  - **Use current timestamp to build log table**: cochez cette case pour utiliser la date et l’heure courantes pour construire la table, pour des raisons d’identification, si vous utilisez plusieurs fois la même table.

### Transfer files secure

Cochez cette case pour transférer le fichier via HTTPS au lieu de HTTP.

### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE_INSERTED</strong></td>
<td>nombre de lignes insérées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong></td>
<td>nombre de lignes supprimées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>FILENAME</strong></td>
<td>nom du fichier traité. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_LOG_FILE</strong></td>
<td>chemin d’accès au fichier local de log. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en standalone.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton <code>[+]</code> pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez <em>Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte</em> à la page 2641 et <em>Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement</em> à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

### Paramètres des différences sources de données importées

Les paramètres de ce composant changent selon la source de votre import de données.

Ce composant peut gérer des données provenant des sources suivantes :

- fichier local
- fichier distant
- base de données EXASol
- base de données Oracle
- base de données compatible JDBC

**Fichier local**
Le fichier local n’est pas transféré lors du chargement du fichier. Le pilote établit un Service Web (sécurisé) envoyant l’URL à la base de données et la base de données récupère le fichier de ce Service Web local. Comme le port de ce Service ne peut être explicitement défini, cette méthode nécessite un réseau transparent entre le Job local Talend et la base de données EXASolution.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File name</td>
<td>Spécifiez le chemin d’accès au fichier local à importer.</td>
</tr>
</tbody>
</table>

**Fichier distant**
Cette méthode fonctionne avec un fichier accessible sur un serveur via les protocoles suivants : SCP, SFTP, FTP, HTTP ou HTTPS.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use predefined connection</td>
<td>Il est possible, via l’interface SQL, de configurer une connexion nommée dans la base de données EXASolution même. Cochez cette case si vous souhaitez utiliser une connexion déjà définie, puis saisissez son nom. Pour savoir quelles sont les connexions disponibles, regardez la table SYS.EXA_DBA_CONNECTIONS dans la base de données. La connexion doit contenir une URL avec l’un des protocoles suivants : SCP, SFTP, FTP, HTTP ou HTTPS. L’URL ne doit pas contenir de nom de fichier. Le nom de fichier est toujours dynamique et doit être fourni par la configuration du composant.</td>
</tr>
<tr>
<td>Remote file server URL</td>
<td>Spécifiez l’URL du serveur de fichiers, sans le nom du fichier lui-même.</td>
</tr>
<tr>
<td>File name</td>
<td>Spécifiez le nom du fichier que vous souhaitez récupérer depuis le serveur.</td>
</tr>
<tr>
<td>Query parameters</td>
<td>Si le Service Web dépend des paramètres de requête, spécifiez ici ces paramètres. Par exemple, si vous souhaitez obtenir un fichier d’un système de fichiers HDFS via le Service Web, vous devez ajouter des paramètres supplémentaires, comme open=true.</td>
</tr>
<tr>
<td>Use user authentication</td>
<td>Cochez cette case si vous souhaitez utiliser l’authentification simple lors de la connexion au serveur Web.</td>
</tr>
<tr>
<td>Remote user et Remote user password</td>
<td>Saisissez l’identifiant et le mot de passe utilisateur nécessaire à l’accès au serveur Web.</td>
</tr>
</tbody>
</table>

**Base de données EXASol**
Une base de données EXASolution peut également servir de source distante pour les données. La source peut être une table ou une requête spécifique.
**Use predefined connection**

Il est possible, via l'interface SQL, de configurer une connexion nommée dans la base de données EXASolution même. Cochez cette case si vous souhaitez utiliser une connexion déjà définie, puis saisissez son nom.

Pour savoir quelles sont les connexions disponibles, regardez la table SYS.EXA_DBA_CONNECTIONS dans la base de données.

L'identifiant et le mot de passe doivent être fournis par le composant et non en tant que partie de la connexion prédéfinie.

**EXASol database host**

Spécifiez l'hôte de la base de données EXASolution distante.

Ce champ peut également être utilisé pour accéder à un cluster.

**Use self defined query**

Cochez cette case pour utiliser une requête spécifique pour obtenir les données.

Cette méthode est conseillée, par exemple, si vos données doivent être filtrées (à l'aide d’une condition **where**), fusionnées ou converties.

**Source query**

Si vous souhaitez utiliser une requête spécifique, saisissez la requête dans ce champ.

**Database or schema**

Si vous n’utilisez pas de requête spécifique, saisissez le nom du schéma de la table source dans ce champ.

**Source table**

Si vous n’utilisez pas de requête spécifique, saisissez le nom de la table dans ce champ.

Le mapping entre les colonnes de la table source et les colonnes de la table cible (colonnes du schéma) peut être configuré dans les paramètres avancés.

**Use user authentication**

Cochez cette case si vous souhaitez utiliser l’authentification simple lors de la connexion à la base de données source.

**Remote user et Remote users password**

Saisissez l’identifiant et le mot de passe utilisateur nécessaire à l’accès à la base de données source.

---

**Base de données Oracle**

Une base de données Oracle peut également servir de source distante pour les données. Accéder à une base de données Oracle nécessite une licence Enterprise pour la base de données EXASolution et ne fonctionne pas avec l’édition gratuite. La source peut être une table ou une requête spécifique.

**Use predefined connection**

Il est possible, via l’interface SQL, de configurer une connexion nommée dans la base de données EXASolution même. Cochez cette case si vous souhaitez utiliser une connexion déjà définie, puis saisissez son nom.

Pour savoir quelles sont les connexions disponibles, regardez la table SYS.EXA_DBA_CONNECTIONS dans la base de données.

L’identifiant et le mot de passe doivent être fournis par le composant et non en tant que partie de la connexion prédéfinie.

**Oracle database URL**

Spécifiez l’URL JDBC pour la base de données Oracle.

**Use self defined query**

Cochez cette case pour utiliser une requête spécifique pour obtenir les données.

Cette méthode est conseillée, par exemple, si vos données doivent être filtrées (à l’aide d’une condition **where**), fusionnées ou converties.
Si vous souhaitez utiliser une requête spécifique, saisissez la requête dans ce champ.

Si vous n’utilisez pas de requête spécifique, saisissez le nom du schéma de la table source dans ce champ.

Si vous n’utilisez pas de requête spécifique, saisissez le nom de la table dans ce champ.

Le mapping entre les colonnes de la table source et les colonnes de la table cible (colonnes du schéma) peut être configuré dans les paramètres avancés.

Cochez cette case si vous souhaitez utiliser l’authentification simple lors de la connexion à la base de données source.

Saisissez l’identifiant et le mot de passe utilisateur nécessaire à l’accès à la base de données source.

Il est possible, via l’interface SQL, de configurer une connexion nommée dans la base de données EXASolution même. Cochez cette case si vous souhaitez utiliser une connexion déjà définie, puis saisissez son nom.

Pour savoir quelles sont les connexions disponibles, regardez la table SYS.EXA_DBA_CONNECTIONS dans la base de données.

L’identifiant et le mot de passe doivent être fournis par le composant et non en tant que partie de la connexion prédéfinie.

Spécifiez l’URL JDBC pour la base de données source.

Cochez cette case pour utiliser une requête spécifique pour obtenir les données.

Cette méthode est conseillée, par exemple, si vos données doivent être filtrées (à l’aide d’une condition where), fusionnées ou converties.

Si vous souhaitez utiliser une requête spécifique, saisissez la requête dans ce champ.

Si vous n’utilisez pas de requête spécifique, saisissez le nom du schéma de la table source dans ce champ.

Si vous n’utilisez pas de requête spécifique, saisissez le nom de la table dans ce champ.

Le mapping entre les colonnes de la table source et les colonnes de la table cible (colonnes du schéma) peut être configuré dans les paramètres avancés.

Cochez cette case si vous souhaitez utiliser l’authentification simple lors de la connexion à la base de données source.

Base de données compatible JDBC

L’édition gratuite de la base de données EXASolution supporte les bases de données MySQL et PostgreSQL et d’autres sont disponibles dans l’édition Enterprise. La source peut être une table ou une requête personnalisée.

Presque toutes les bases de données de niveau entreprise fournissent une interface JDBC.

Il est possible, via l’interface SQL, de configurer une connexion nommée dans la base de données EXASolution même. Cochez cette case si vous souhaitez utiliser une connexion déjà définie, puis saisissez son nom.

Pour savoir quelles sont les connexions disponibles, regardez la table SYS.EXA_DBA_CONNECTIONS dans la base de données.

L’identifiant et le mot de passe doivent être fournis par le composant et non en tant que partie de la connexion prédéfinie.

Spécifiez l’URL JDBC pour la base de données source.

Cochez cette case pour utiliser une requête spécifique pour obtenir les données.

Cette méthode est conseillée, par exemple, si vos données doivent être filtrées (à l’aide d’une condition where), fusionnées ou converties.

Si vous souhaitez utiliser une requête spécifique, saisissez la requête dans ce champ.

Si vous n’utilisez pas de requête spécifique, saisissez le nom du schéma de la table source dans ce champ.

Si vous n’utilisez pas de requête spécifique, saisissez le nom de la table dans ce champ.

Le mapping entre les colonnes de la table source et les colonnes de la table cible (colonnes du schéma) peut être configuré dans les paramètres avancés.

Cochez cette case si vous souhaitez utiliser l’authentification simple lors de la connexion à la base de données source.
**Scénario : Importer des données dans une table de base de données EXASolution à partir d’un fichier local CSV**

Ce scénario décrit un Job écrivant des informations relatives à des employés dans un fichier CSV, puis chargeant les données de ce fichier local dans une nouvelle table de base de données crée dans une base de données EXASolution, à l’aide du composant **tEXABulkExec**, puis récupérant les données de la table et les affichant dans la console.

**Construire le Job**

**Procédure**

1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette : un **tFixedFlowInput**, un **tFileOutputDelimited**, un **tEXABulkExec**, un **tEXAInput** et un **tLogRow**.
2. Reliez le **tFixedFlowInput** au **tFileOutputDelimited** à l’aide d’un lien Row > Main.
3. Répétez l’opération pour connecter le **tEXAInput** au **tLogRow**.
4. Connectez le **tFixedFlowInput** au composant **tEXABulkExec** à l’aide d’un lien Trigger > OnSubjobOk.
5. Reliez le **tEXABulkExec** au **tEXAInput** de la même manière.
**Configurer les composants**

**Préparer les données source**

**Procédure**

1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton [...] à côté de **Edit schema** pour ouvrir la boîte de dialogue [Schema].

3. Cliquez sur le bouton [+] pour ajouter six colonnes : **EmployeeID** de type **Integer**, **EmployeeName**, **OrgTeam** et **JobTitle** de type **String**, **OnboardDate** de type **Data** avec le modèle de date **yyyy-MM-dd** et **MonthSalary** de type **Double**.

5. Dans la zone **Mode**, sélectionnez **Use Inline Content (delimited file)** et saisissez les données employés suivantes dans le champ **Content**.

<table>
<thead>
<tr>
<th>ID</th>
<th>Nom</th>
<th>Groupe</th>
<th>Poste</th>
<th>Date de naissance</th>
<th>Salaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>12000</td>
<td>James</td>
<td>Dev Team</td>
<td>Developer</td>
<td>2008-01-01</td>
<td>15000.01</td>
</tr>
<tr>
<td>12001</td>
<td>Jimmy</td>
<td>Dev Team</td>
<td>Developer</td>
<td>2008-11-22</td>
<td>13000.11</td>
</tr>
<tr>
<td>12002</td>
<td>Herbert</td>
<td>QA Team</td>
<td>Tester</td>
<td>2008-05-12</td>
<td>12000.22</td>
</tr>
<tr>
<td>12003</td>
<td>Harry</td>
<td>Doc Team</td>
<td>Technical Writer</td>
<td>2009-03-10</td>
<td>12000.33</td>
</tr>
<tr>
<td>12004</td>
<td>Ronald</td>
<td>QA Team</td>
<td>Tester</td>
<td>2009-06-20</td>
<td>12500.44</td>
</tr>
<tr>
<td>12005</td>
<td>Mike</td>
<td>Dev Team</td>
<td>Developer</td>
<td>2009-10-15</td>
<td>14000.55</td>
</tr>
<tr>
<td>12006</td>
<td>Jack</td>
<td>QA Team</td>
<td>Tester</td>
<td>2009-03-25</td>
<td>13500.66</td>
</tr>
<tr>
<td>12007</td>
<td>Thomas</td>
<td>Dev Team</td>
<td>Developer</td>
<td>2010-02-20</td>
<td>16000.77</td>
</tr>
<tr>
<td>12008</td>
<td>Michael</td>
<td>Dev Team</td>
<td>Developer</td>
<td>2010-07-15</td>
<td>14000.88</td>
</tr>
<tr>
<td>12009</td>
<td>Peter</td>
<td>Doc Team</td>
<td>Technical Writer</td>
<td>2011-02-10</td>
<td>12500.99</td>
</tr>
</tbody>
</table>

6. Double-cliquez sur le composant **tFileOutputDelimited** pour ouvrir sa vue **Basic settings**.

7. Dans le champ **File Name**, spécifiez le fichier dans lequel les données d'entrée seront écrites. Dans cet exemple, le fichier est "E:/employee.csv".

8. Cliquez sur **Advanced settings** pour ouvrir la vue **Advanced settings** du **tFileOutputDelimited**.

9. Cochez la case **Advanced separator (for numbers)** et, dans les champs **Thousands separator** et **Decimal separator** qui s'affichent, spécifiez le séparateurs des milliers et des nombres décimaux. Dans cet exemple, les valeurs par défaut ",” et "." sont utilisées.

**Charger les données source dans une nouvelle table de base de données EXASolution**

**Procédure**

1. Double-cliquez sur le composant **tEXABulkExec** pour ouvrir sa vue **Basic settings**.
2. Renseignez les champs **Host**, **Port**, **Schema**, **User** et **Password** avec vos informations de connexion à la base de données EXASolution.

3. Dans le champ **Table**, saisissez le nom de la table dans laquelle les données source seront écrites. Dans cet exemple, la table cible est nommée "employee" et n’existe pas encore.

4. Sélectionnez **Create table** dans la liste **Action on table** afin de créer la table spécifiée.

5. Dans la zone **Source**, sélectionnez **Local Filename** comme source pour les données d’entrée et spécifiez le fichier contenant les données source. Dans cet exemple, le fichier est "E:/employee.csv".

6. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir la boîte de dialogue [Schema] et définir le schéma, qui doit être le même que celui de composant **tFixedFlowInput**. Cliquez sur **OK** afin de valider ces modifications et fermer la boîte de dialogue.

7. Cliquez sur **Advanced settings** pour ouvrir la vue **Advanced settings** du composant **tEXABulkExec**.

8. Dans la table **Column Formats**, pour les deux champs numériques, **EmployeeID** et **MonthSalary**, cochez la case correspondante dans la colonne **Has Thousand Delimiters** et définissez la chaîne de...

9. Assurez-vous que les champs Thousands Separator et Decimal Separator ont une valeur identique à celle dans le composant tFileOutputDelimited et laissez les autres paramètres par défaut.

Récupérer les données de la table EXASolution

Procédure

1. Double-cliquez sur le composant tEXAInput pour ouvrir sa vue Basic settings.

2. Renseignez les champs Host name, Port, Schema name, Username et Password avec vos informations de connexion à la base de données EXASolution.

3. Dans le champ Table Name, saisissez le nom de la table de laquelle récupérer les données. Dans cet exemple, la table est “employee”.


   Cliquez sur OK pour fermer la boîte de dialogue et accepter la propagation du schéma au composant suivant.

5. Cliquez sur le bouton Guess Query pour renseigner le champ Query avec l'instruction SQL auto-générée à exécuter sur la table spécifiée.

   ```
   SELECT employee.EmployeeID,
   employee.EmployeeName,
   employee.OrgTeam,
   employee.JobTitle,
   employee.OnboardDate,
   employee.MonthSalary
   FROM employee
   ```

6. Double-cliquez sur le composant tLogRow pour ouvrir sa vue Basic settings.
7. Dans la zone **Mode**, sélectionnez l’option **Table (print values in cells of a table)** pour une lisibilité optimale de la sortie.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** pour l’exécuter.

```plaintext
[statistics] connecting to socket on port 3680
[statistics] connected

<table>
<thead>
<tr>
<th>EmployeeID</th>
<th>EmployeeName</th>
<th>OrgTeam</th>
<th>JobTitle</th>
<th>OnboardDate</th>
<th>MonthSalary</th>
</tr>
</thead>
<tbody>
<tr>
<td>12000</td>
<td>James</td>
<td>Dev Team</td>
<td>Developer</td>
<td>2008-01-01</td>
<td>15000.01</td>
</tr>
<tr>
<td>12001</td>
<td>Jimmey</td>
<td>Dev Team</td>
<td>Developer</td>
<td>2008-11-22</td>
<td>13000.11</td>
</tr>
<tr>
<td>12002</td>
<td>Herbert</td>
<td>CA Team</td>
<td>Tester</td>
<td>2008-05-12</td>
<td>12000.22</td>
</tr>
<tr>
<td>12003</td>
<td>Harry</td>
<td>Doc Team</td>
<td>Technical Writer</td>
<td>2009-03-10</td>
<td>12000.33</td>
</tr>
<tr>
<td>12004</td>
<td>Ronald</td>
<td>CA Team</td>
<td>Tester</td>
<td>2009-06-20</td>
<td>12500.44</td>
</tr>
<tr>
<td>12005</td>
<td>Mike</td>
<td>Dev Team</td>
<td>Developer</td>
<td>2009-10-15</td>
<td>14000.55</td>
</tr>
<tr>
<td>12006</td>
<td>Jack</td>
<td>CA Team</td>
<td>Tester</td>
<td>2009-03-25</td>
<td>13500.66</td>
</tr>
<tr>
<td>12007</td>
<td>Thomas</td>
<td>Dev Team</td>
<td>Developer</td>
<td>2010-02-20</td>
<td>16000.77</td>
</tr>
<tr>
<td>12008</td>
<td>Michael</td>
<td>Dev Team</td>
<td>Developer</td>
<td>2010-07-15</td>
<td>14000.88</td>
</tr>
<tr>
<td>12009</td>
<td>Peter</td>
<td>Doc Team</td>
<td>Technical Writer</td>
<td>2011-02-10</td>
<td>12500.99</td>
</tr>
</tbody>
</table>
```

[statistics] disconnected

Comme indiqué dans la capture d’écran, les données des employés sont écrites dans la table de la base de données EXASolution spécifiée et sont récupérées et affichées dans la console.
tEXAClose

Ce composant ferme une connexion active à une instruction de base de données EXASolution pour libérer les ressources utilisées.

Propriétés du tEXAClose Standard

Ces propriétés sont utilisées pour configurer le tEXAClose s'exécutant dans le framework de Jobs Standard.

Le composant tEXAClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Component List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dans la liste, sélectionnez le composant tEXAConnection qui ouvre la connexion que vous devez fermer.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en association avec des composants EXASolution, notamment avec le tEXAConnection et le tEXACommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tEXACCommit

Ce composant valide les données traitées dans le Job dans la base de données EXASolution connectée.

Le tEXACCommit utilise une connexion unique pour commite en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

Propriétés du tEXACCommit Standard

Ces propriétés sont utilisées pour configurer le tEXACCommit s’exécutant dans le framework de Jobs Standard.

Le composant tEXACCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component List</td>
<td>Sélectionnez le composant tEXAConnection via lequel vous souhaitez effectuer un commit.</td>
</tr>
</tbody>
</table>
| Close Connection                | Cette case est cochée par défaut et vous permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. Avertissement :

Si vous souhaitez utiliser un lien Row > Main pour relier le tEXACCommit à votre Job, vous données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close Connection car la connexion sera fermée avant la fin du commit de votre première ligne. |

Advanced settings

| tStatCatcher Statistics         | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |
### Global Variables

| **Global Variables** | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |
| --- | --- |

### Utilisation

<table>
<thead>
<tr>
<th><strong>Règle d’utilisation</strong></th>
<th>Ce composant est généralement utilisé avec des composants EXASolution, notamment le tEXAConnection et le tEXARollback.</th>
</tr>
</thead>
</table>
| **Dynamic settings** | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Scénario associé

Pour un scénario similaire utilisant une autre base de données, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tEXAConnection

Ce composant ouvre une connexion à une instance de la base de données EXASolution pouvant être réutilisée par d’autres composants EXASolution.

Propriétés du tEXAConnection Standard

Ces propriétés sont utilisées pour configurer le tEXAConnection s’exécutant dans le framework de Jobs Standard.

Le composant tEXAConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property Type</td>
<td>Peut être Built-In ou Repository.</td>
</tr>
<tr>
<td></td>
<td>• Built-In : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>• Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Saisissez l’hôte ou la liste d’hôtes des serveurs de bases de données EXASol. EXASol peut s’exécuter dans un environnement de cluster. Une valeur valide peut être une adresse IP simple (par exemple 172.16.173.128), une liste d’adresses IP (par exemple 172.16.173.128..130 qui représente trois serveurs 172.16.173.128, 172.16.173.129 , et 172.16.173.130), ou une liste d’hôtes séparés par des virgules (par exemple server1,server2,server3) du cluster de base de données EXASolution.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute du cluster de base de données EXASolution.</td>
</tr>
<tr>
<td>Schema</td>
<td>Saisissez le nom du schéma que vous souhaitez utiliser.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les données d’authentification de l’utilisateur pour accéder à la base de données EXASolution. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue</td>
</tr>
</tbody>
</table>
qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Use or register a shared DB Connection**

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un composant **tRunJob** ayant une de ces options activée fera échouer votre Job.

### Advanced settings

#### Auto Commit

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commit pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

#### Additional JDBC Parameters

Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, `encryption=1;clientname=Talend`.

Ce champ n’est pas disponible si la case **Use an existing connection** est cochée.

#### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.
Global Variables

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé avec des composants EXASolution, notamment le **tEXACommit** et le **tEXARollback**.

**Scénario associé**

Pour un scénario similaire utilisant une autre base de données, consultez *Scénario : Insérer des données dans des tables mère/fille* à la page 2620.
tEXAInput

Ce composant récupère des données d’une base de données EXASolution, à l’aide d’une requête, en respectant un ordre défini correspondant à la définition du schéma, puis les passe au composant suivant.

Propriétés du tEXAInput Standard

Ces propriétés sont utilisées pour configurer le tEXAInput s’exécutant dans le framework de Jobs Standard.

Le composant tEXAInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
</table>
| Property Type | Peut être Built-In ou Repository.  
|              | • Built-In : Propriétés utilisées ponctuellement.  
|              | • Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |

Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.

Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.

Use an existing connection

Cochez cette case et, dans la liste qui s’affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

**Host name**
Saisissez l’hôte ou la liste d’hôtes des serveurs de bases de données EXASol. EXASol peut s’exécuter dans un environnement de cluster. Une valeur valide peut être une adresse IP simple (par exemple 172.16.173.128), une liste d’adresses IP (par exemple 172.16.173.128..130 qui représente trois serveurs 172.16.173.128, 172.16.173.129 et 172.16.173.130), ou une liste d’hôtes séparés par des virgules (par exemple **server1,server2,server3**) du cluster de base de données EXASolution.

**Port**
Saisissez le numéro du port d’écoute du cluster de base de données EXASolution.

**Schema name**
Saisissez le nom du schéma que vous souhaitez utiliser.

**Username et Password**
Saisissez les données d’authentification de l’utilisateur pour accéder à la base de données EXASolution.

Pour saisir le mot de passe, cliquez sur le bouton [..] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Schema et Edit schema**
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.
- **Repository** : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
**Table Name**

Saisissez le nom de la table.

**Query Type et Query**

Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.

**Guess Query**

Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.

**Guess schema**

Cliquez sur le bouton pour récupérer le schéma de la table.

### Advanced settings

**Change fetch size**

Cochez cette case pour modifier la taille d'extraction qui définit le volume de données de résultat envoyé pour une étape de communication avec la base de données. Dans le champ **Fetch size**, vous devez saisir la taille en kilo-octets.

**Additional JDBC parameters**

Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, 

`encryption=1;clientname=Talend`.

Ce champ n'est pas disponible si la case **Use an existing connection** est cochée.

**Trim all the String/Char columns**

Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.

**Trim column**

Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.

Cette option n'est pas disponible si la case **Trim all the String/Char columns** est cochée.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.
**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QUERY : requête traitée. Cette variable est une variable <strong>Flow</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en tant que composant de départ pour un Job ou un sous-job et il doit être relié par un lien de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable.</td>
</tr>
<tr>
<td></td>
<td>Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les</td>
</tr>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez Scénario : Importer des données dans une table de base de données EXASolution à partir d’un fichier local CSV à la page 939.

Pour des scénarios similaires utilisant d’autres bases de données, consultez : 

paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.
tEXAOOutput

Ce composant écrit, met à jour, modifie ou supprime des données dans une base de données EXASolution en exécutant l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Propriétés du tEXAOOutput Standard

Ces propriétés sont utilisées pour configurer le tEXAOOutput s’exécutant dans le framework de Jobs Standard.

Le composant tEXAOOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Peut être Built-In ou Repository.</td>
</tr>
<tr>
<td></td>
<td>• Built-In : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>• Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><img src="icon" alt="Icone de connexion" /></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et, dans la liste qui s’affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.</td>
</tr>
<tr>
<td></td>
<td>Remarque :</td>
</tr>
</tbody>
</table>
| | Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Host</th>
<th>Saisissez l’hôte ou la liste d’hôtes des serveurs de bases de données EXASol. EXASol peut s’exécuter dans un environnement de cluster. Une valeur valide peut être une adresse IP simple (par exemple 172.16.173.128), une liste d’adresses IP (par exemple 172.16.173.128..130 qui représente trois serveurs 172.16.173.128, 172.16.173.129, et 172.16.173.130), ou une liste d’hôtes séparés par des virgules (par exemple server1,server2,server3) du cluster de base de données EXASolution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute du cluster de base de données EXASolution.</td>
</tr>
<tr>
<td>Schema name</td>
<td>Saisissez le nom du schéma que vous souhaitez utiliser.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les données d’authentification de l’utilisateur pour accéder à la base de données EXASolution. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :

  - **None** : n’effectuer aucune opération de table.
  - **Drop and create table** : supprimer la table puis en créer une nouvelle.
  - **Create table** : créer une table qui n’existe pas encore.
  - **Create table if does not exist** : créer la table si nécessaire.
  - **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.
  - **Clear table** : supprimer le contenu de la table.
  - **Truncate table** : supprimer le contenu de la table, sans possibilité de rollback. |
Action on data

Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

- **Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
- **Update** : Mettre à jour les entrées existantes.
- **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
- **Delete** : Supprimer les entrées correspondantes au flux d’entrée.

**Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Die on error

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.  
Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.  
Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Use commit control</strong></th>
<th>Cochez cette case afin de définir, dans le champ <strong>Commit every</strong>, qui apparaît alors, le nombre de lignes à traiter avant de commiter.</th>
</tr>
</thead>
</table>
| **Additional JDBC parameters** | Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, **encryption=1;clientname=Talend**.  
Ce champ n’est pas disponible si la case **Use an existing connection** est cochée. |
| **Additional Columns** | Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.  
- **Name** : Saisissez le nom de la colonne à modifier ou à insérer.  
- **SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.  
- **Position** : Sélectionnez **Before, Replace** ou **After**, en fonction de l’action à effectuer sur la colonne de référence. |
- **Reference column**: Saisissez une colonne de référence pour situer ou remplacer la nouvelle colonne ou celle à modifier.

## Use field options

- **Key in update**: Cochez cette case dans la colonne sur laquelle se base la mise à jour des données.
- **Key in delete**: Cochez cette case dans la colonne sur laquelle se base la suppression des données.
- **Updatable**: Cochez cette case si les données de la colonne peuvent être mises à jour.
- **Insertable**: Cochez cette case si les données de la colonne peuvent être insérées.

## Enable debug mode

Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.

## Use batch mode

Cochez cette case pour activer le mode de traitement par lots pour le traitement des données et dans le champ **Batch Size**, saisissez le nombre d’enregistrements à effectuer dans chaque lot.

## tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong></td>
<td>nombre de lignes mises à jour. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong></td>
<td>nombre de lignes insérées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong></td>
<td>nombre de lignes supprimées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_REJECTED</strong></td>
<td>nombre de lignes rejetées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette
Il permet de faire des actions sur une table ou les données d’une table d’une base de données EXA. Il permet aussi de créer un flux de rejet avec un lien **Row > Rejects** filtrant les données en erreur. Pour un exemple d’utilisation, consultez **Scénario : Récupérer les données erronées à l’aide d’un lien Reject** à la page 2675 du composant **tMysqlOutput**.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

**Limitation**

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center ([https://help.talend.com](https://help.talend.com)).
Scénario associé

Pour des scénarios utilisant d'autres bases de données, consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667.
tEXARollback

Ce composant annule le commit de transaction dans la base de données EXASolution.
Le tEXARollback vous permet d’effectuer un rollback sur tout changement effectué dans la base de données EXASolution et ainsi d’éviter le commit de transaction involontaire.

Propriétés du tEXARollback Standard

Ces propriétés sont utilisées pour configurer le tEXARollback s’exécutant dans le framework de Jobs Standard.
Le composant tEXARollback Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component List</td>
<td>Sélectionnez le composant tEXAConnection sur lequel vous souhaitez effectuer un rollback.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Cette case est cochée par défaut et vous permet de fermer la connexion à la base de données une fois le rollback effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics   | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Global Variables

| Global Variables          | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en association avec des composants EXASolution, notamment avec le <em>tEXAConnection</em> et le <em>tEXACommit</em>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton <code>[+]</code> pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour voir un scénario similaire utilisant une autre base de données, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623.
tEXARow

Ce composant exécute des requêtes SQL sur une base de données EXASolution.

Le tEXARow, selon la nature de la requête et de la base de données, agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tEXARow Standard

Ces propriétés sont utilisées pour configurer le tEXARow s’exécutant dans le framework de Jobs Standard.

Le composant tEXARow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Peut être Built-In ou Repository.</td>
</tr>
<tr>
<td></td>
<td>• Built-In : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>• Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et, dans la liste qui s’affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.</td>
</tr>
</tbody>
</table>

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

| **Host** | Saisissez l'hôte ou la liste d'hôtes des serveurs de bases de données EXASol. EXASol peut s'exécuter dans un environnement de cluster. Une valeur valide peut être une adresse IP simple (par exemple 172.16.173.128), une liste d'adresses IP (par exemple 172.16.173.128..130 qui représente trois serveurs 172.16.173.128, 172.16.173.129, et 172.16.173.130), ou une liste d'hôtes séparés par des virgules (par exemple server1,server2,server3) du cluster de base de données EXASolution. |
| **Port** | Saisissez le numéro du port d'écoute du cluster de base de données EXASolution. |
| **Schema name** | Saisissez le nom du schéma que vous souhaitez utiliser. |
| **Username et Password** | Saisissez les données d'authentification de l'utilisateur pour accéder à la base de données EXASolution. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez |

---
**Table Name**
Nom de la table de base de données à modifier.

**Query Type**
Peut être **Built-In** ou **Repository**.
- **Built-In** : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.
- **Repository** : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.

**Guess Query**
Cliquez sur le bouton **Guess Query** pour générer la requête qui correspond au schéma de la table dans le champ **Query**.

**Query**
Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.

**Die on error**
Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.
Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.

---

**Advanced settings**

**Additional JDBC parameters**
Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, `encryption=1;clientname=Talend`.
Ce champ n’est pas disponible si la case **Use an existing connection** est cochée.

**Propagate QUERY’s recordset**
Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.

**Use PreparedStatement**
Cochez cette case pour utiliser des instructions préparées et dans le tableau **Set PreparedStatement Parameters**, ajoutez autant de paramètres que nécessaire puis définissez les attributs suivants pour chaque paramètre :
- **Parameter Index** : Saisissez l’index du paramètre d’instruction préparée.
- **Parameter Type** : Cliquez dans la cellule et sélectionnez le type de paramètre dans la liste.
- **Parameter Value** : Saisissez la valeur du paramètre.
Commit every

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

TStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

Global Variables

QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.

Limitation


Scénario associé

Pour voir des scénarios similaires utilisant d’autres bases de données, consultez :

• Procédure,
• Scénario : Supprimer et re-générer un index de table MySQL à la page 2700.
tEXistConnection

Ce composant ouvre une connexion à une base de données eXist afin d’effectuer une transaction.

Propriétés du tEXistConnection Standard

Ces propriétés sont utilisées pour configurer le tEXistConnection s’exécutant dans le framework de Jobs Standard.
Le composant tEXistConnection Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>URI</th>
<th>URI de la base de données à laquelle vous souhaitez vous connecter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection</td>
<td>Saisissez le chemin d’accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.</td>
</tr>
<tr>
<td>Driver</td>
<td>Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s’affiche.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

| Règle d’utilisation | Ce composant est étroitement lié aux tEXistGet et tEXistPut. Une fois que vous avez défini les propriétés de connexion du composant, vous pouvez réutiliser cette connexion sans avoir à définir à nouveau les propriétés pour chaque composant tEXist utilisé dans le Job. La base de données eXist est un système de gestion de base de données open source, construit à l’aide de la technologie XML. Elle stocke des données XML selon le |

Limitation
Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d'informations concernant l'installation des modules externes dans Talend Help Center (https://help.talend.com).

**Scénario associé**

Pour un scénario associé au **tEXistConnection**, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620 du composant **tMysqlConnection**.
tEXistDelete

Ce composant supprime des ressources spécifiées d’une base de données eXist distante.

Propriétés du tEXistDelete Standard

Ces propriétés sont utilisées pour configurer le tEXistDelete s’exécutant dans le framework de Jobs Standard.
Le composant tEXistDelete Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

| Use an existing connection/Component List | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.
Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>URI de la base de données à laquelle vous souhaitez vous connecter.</td>
</tr>
<tr>
<td>Collection</td>
<td>Saisissez le chemin d’accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.</td>
</tr>
<tr>
<td>Driver</td>
<td>Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s’affiche.</td>
</tr>
<tr>
<td><strong>Remarque</strong></td>
<td>Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Target Type</td>
<td>Sélectionnez le type de cible : Resource(Ressource), Collection(Collection), ou All(Tous).</td>
</tr>
<tr>
<td>Files</td>
<td>Cliquez sur le bouton [+] afin d’ajouter les lignes que vous souhaitez utiliser comme filtres :</td>
</tr>
</tbody>
</table>
**Filemask** : saisissez le nom du fichier ou le masque de fichier, en utilisant des caractères de remplacement (*) ou des expressions régulières.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable After et retourne un entier. **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**. |

### Utilisation


| Limitation | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center ([https://help.talend.com](https://help.talend.com)). |
Scénario associé

Aucun scénario n'est disponible pour la version Standard de ce composant.
**tEXistGet**

Ce composant récupère les ressources sélectionnées d’une base de données eXist distante et les met dans un répertoire local spécifié.

**Propriétés du tEXistGet Standard**

Ces propriétés sont utilisées pour configurer le tEXistGet s’exécutant dans le framework de Jobs Standard.

Le composant tEXistGet Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection/Component List</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.</td>
</tr>
<tr>
<td>URI</td>
<td>URI de la base de données à laquelle vous souhaitez vous connecter.</td>
</tr>
<tr>
<td>Collection</td>
<td>Saisissez le chemin d’accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.</td>
</tr>
<tr>
<td>Driver</td>
<td>Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s’affiche.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong> Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [••] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Local directory</td>
<td>Chemin d’accès au fichier de destination.</td>
</tr>
<tr>
<td>Files</td>
<td>Cliquez sur le bouton [+] afin d’ajouter les lignes que vous souhaitez utiliser comme filtres : Filemask : saisissez le nom du fichier ou le masque de fichier, en utilisant des caractères de remplacement (*) ou des expressions régulières.</td>
</tr>
</tbody>
</table>
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | **NB.FILE** : Indique le nombre de fichiers traités. Cette variable est une variable **After** et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant est utilisé en tant que sous-job ou comme composant de sortie. La base de données eXist est un système de gestion de base de données open source, construit à l’aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l’index. Pour plus d’informations concernant XQuery, consultez : XQuery (en anglais). Pour plus d’informations concernant les extensions de XQuery Update, consultez : XQuery update extension |
| Limitation | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com). |
Scénario : Récupérer des ressources à partir d'un serveur distant de base de données eXist

Ce scénario décrit un Job à un composant qui permet de récupérer des données à partir d'un serveur de base de données eXist et de télécharger les données dans un répertoire local.

Ce Job simple nécessite un composant : le tEXistGet.

Procédure

1. A partir de la Palette, déposez le composant tEXistGet dans l'espace de modélisation graphique.
2. Double-cliquez sur le composant tEXistGet pour définir ses propriétés dans l'onglet Basic settings de sa vue Component.
3. Renseignez le champ URI en saisissant l'adresse URI de la base de données eXist à laquelle vous souhaitez vous connecter.
   Dans ce scénario, l'URI est `xmldb:exist://192.168.0.165:8080/exist/xmlrpc`. Notez que cette adresse URI est destinée essentiellement à la démonstration et que ce n'est pas une adresse valide.
4. Remplissez le champ Collection avec le chemin d'accès à la collection souhaitée sur le serveur de la base de données, `/db/talend` dans ce scénario.
5. Renseignez le champ Driver en saisissant le pilote de la base de données XML, `org.exist.xmldb.DatabaseImpl` dans ce scénario.
6. Remplissez les champs **Username** et **Password** avec *admin* et *talend* respectivement pour ce scénario.

7. Cliquez sur le bouton [...] à côté du champ **Local directory** pour définir le chemin d'accès au répertoire dans lequel le fichier XML téléchargé à partir du serveur de la base de données sera sauvegardé.

Dans ce scénario, faites pointer ce chemin vers votre bureau, par exemple : 
*C:/Documents and Settings/galano/Desktop/ExistGet.*

8. Dans le champ **Files**, cliquez sur le bouton [+] afin d'ajouter une nouvelle ligne dans la zone **Filemask** et remplissez-la avec un nom de fichier complet pour récupérer les données à partir d’un fichier spécifique sur le serveur, ou avec un masque de fichier (**filemask**) pour récupérer les données à partir d’un ensemble de fichiers. Dans ce scénario, saisissez *dictionary_en.xml*.


**Résultats**

```xml
<!-- generated by ToXgene Version 1.1a in Wed Jun 23 12:16:00 EDT 2004 -->
<dictionary>
  <e id="E1">
    <hwr>
      <hwr>planks</hwr>
      <pr>!gknl+J]i8u</pr>
    </hwr>
    <hwr>gaul</hwr>
    <pr>Zft"OeKY</pr>
    <pos>adv.</pos>
  </hwr>
  <et>
    <cr>E709</cr>
    <cr>E3</cr>
    <cr>E439</cr>
    <cr>E414</cr>
  </et>
  <ss>
    <s>
      <def>forges doubt ironic, sly dugouts:sly, regular patt
    </def>
    <qp>
      <q>
        <qd>259</qd>
        <w>closely brave</w>
        <loc>Vh6f~VtSc[</loc>
      </q>
    </qp>
  </ss>
</dictionary>
```

Le fichier XML *dictionary_en.xml* est récupéré et téléchargé dans le répertoire local défini préalablement.
tEXistList

Ce composant liste les ressources stockées dans une base de données eXist distante.

**Propriétés du tEXistList Standard**

Ces propriétés sont utilisées pour configurer le tEXistList s’exécutant dans le framework de Jobs Standard.

Le composant tEXistList Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use an existing connection/Component List</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.</td>
</tr>
<tr>
<td><strong>URI</strong></td>
<td>URI de la base de données à laquelle vous souhaitez vous connecter.</td>
</tr>
<tr>
<td><strong>Collection</strong></td>
<td>Saisissez le chemin d’accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Driver</strong></td>
<td>Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s’affiche. <strong>Remarque :</strong> Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Files</strong></td>
<td>Cliquez sur le bouton [+] afin d’ajouter les lignes que vous souhaitez utiliser comme filtres : <strong>Filemask</strong> : saisissez le nom du fichier ou le masque de fichier, en utilisant des caractères de remplacement (*) ou des expressions régulières.</td>
</tr>
<tr>
<td><strong>Target Type</strong></td>
<td>Sélectionnez le type de cible :</td>
</tr>
</tbody>
</table>
**Resource (Ressource), Collection (Collection), ou All (Tous).**

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Global Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_FILE</strong> : nombre de fichiers itérés. Cette variable est une variable <em>After</em> et retourne un nombre entier.</td>
<td></td>
</tr>
<tr>
<td><strong>CURRENT_FILE</strong> : nom du fichier courant. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
<td></td>
</tr>
<tr>
<td><strong>CURRENT_FILEPATH</strong> : nom du fichier courant ainsi que son chemin d'accès. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
<td></td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option.</td>
<td></td>
</tr>
</tbody>
</table>

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec un *tEXistGet* afin de récupérer les fichiers listés, par exemple. La base de données eXist est un système de gestion de base de données open source, construit à l’aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l’index. Pour plus d’informations concernant XQuery, consultez : *XQuery* (en anglais). Pour plus d’informations concernant les extensions de XQuery Update, consultez : [http://exist-db.org/exist/apps/doc/update_ext.xml](http://exist-db.org/exist/apps/doc/update_ext.xml). |

| Limitation | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton *Install* dans |
Scénario associé

Pour un scénario associé au tEXistList, consultez Lister et obtenir des fichiers/dossiers d’un répertoire FTP à la page 1299.
**tEXistPut**

Ce composant charge des fichiers spécifiés d'un répertoire local défini dans une base de données eXist distante.

**Propriétés du tEXistPut Standard**

Ces propriétés sont utilisées pour configurer le tEXistPut s'exécutant dans le framework de Jobs Standard.

Le composant tEXistPut Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Use an existing connection/Component List | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.

Notez que lorsqu'un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>URI de la base de données à laquelle vous souhaitez vous connecter.</td>
</tr>
<tr>
<td>Collection</td>
<td>Saisissez le chemin d'accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.</td>
</tr>
</tbody>
</table>
| Driver                                 | Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s'affiche.

**Remarque :**

Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins. |
| Username et Password                    | Informations d'authentification de l'utilisateur de base de données. 

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Local directory                         | Chemin d'accès au fichier de destination. |
| Files                                   | Cliquez sur le bouton [+] afin d'ajouter les lignes que vous souhaitez utiliser comme filtres :

**Filemask** : saisissez le nom du fichier ou le masque de fichier, en utilisant des caractères de remplacement (*) ou des expressions régulières. |
Advanced settings

**tStatCatcher Statistics**

| Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables

**Global Variables**

| NB_FILE : Indique le nombre de fichiers traités. Cette variable est une variable *After* et retourne un entier. |
| **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. |
| Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. |
| Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espacement** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
| Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

Utilisation

**Règle d’utilisation**

| Ce composant est utilisé en tant que sous-job ou comme composant de sortie. La base de données eXist est un système de gestion de base de données open source, construit à l’aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l’index. Pour plus d’informations concernant XQuery, consultez : **XQuery (en anglais)**. Pour plus d’informations concernant les extensions de XQuery Update, consultez : [http://exist-db.org/exist/apps/doc/update_ext.xml](http://exist-db.org/exist/apps/doc/update_ext.xml) |

**Limitation**


Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tEXistXQuery

Ce composant interroge des fichiers XML situés dans des bases de données distantes à l’aide de fichiers locaux contenant des requêtes XPath et écrit les résultats en sortie dans un fichier XML stocké localement.

Propriétés du tEXistXQuery Standard

Ces propriétés sont utilisées pour configurer le tEXistXQuery s’exécutant dans le framework de Jobs Standard.

Le composant tEXistXQuery Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection/Component List</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>URI de la base de données à laquelle vous souhaitez vous connecter.</td>
</tr>
<tr>
<td>Collection</td>
<td>Saisissez le chemin d’accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données.</td>
</tr>
</tbody>
</table>
| Driver                                  | Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s’affiche.  

**Remarque:** Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins. |
| Username et Password                    | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| XQuery Input File                       | Parcourez votre répertoire jusqu’au fichier local contenant la requête à exécuter. |
| Local Output                            | Parcourez votre répertoire jusqu’à l’emplacement où les résultats de la requête doivent être sauvegardés. |
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_FILE : Indique le nombre de fichiers traités. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est utilisé en tant que composant unique d’un Job mais peut également être utilisé dans un Job plus complexe. La base de données eXist est un système de gestion de base de données open source, construit à l’aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l’index. Pour plus d’informations concernant XQuery, consultez : <a href="https://www.w3.org/TR/xquery">XQuery (en anglais)</a>. Pour plus d’informations concernant les extensions de XQuery Update, consultez : <a href="https://www.w3.org/TR/xquery-update">XQuery update extension</a>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tEXistXUpdate

Ce composant traite des enregistrements XML et met à jour les enregistrements existants sur le serveur de base de données.

Propriétés du tEXistXUpdate Standard

Ces propriétés sont utilisées pour configurer le tEXistXUpdate s’exécutant dans le framework de Jobs Standard.
Le composant tEXistXUpdate Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

| Use an existing connection/Component List | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.
Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
| URI | URI de la base de données à laquelle vous souhaitez vous connecter. |
| Collection | Saisissez le chemin d’accès à la collection à laquelle vous souhaitez accéder, sur le serveur de la base de données. |
| Driver | Ce champ est automatiquement rempli. Par défaut, le nom du pilote standard s’affiche. |
| Username et Password | Informations d’authentification de l’utilisateur de base de données. |
| Update File | Parcourez votre répertoire jusqu’au fichier local à utiliser pour mettre à jour les enregistrements de la base de données. |

Remarque :
Les utilisateurs peuvent spécifier un pilote différent, selon leurs besoins.
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable *After* et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  
Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

### Utilisation

| Règle d’utilisation | Ce composant est utilisé en tant que composant unique d’un Job mais peut également être utilisé dans un Job plus complexe. La base de données eXist est un système de gestion de base de données open source, construit à l’aide de la technologie XML. Elle stocke des données XML selon le modèle de données XML et offre une fonction efficace de traitement de la requête XQuery, basée sur l’index. Pour plus d’informations concernant XQuery, consultez : [XQuery](en anglais). Pour plus d’informations concernant les extensions de XQuery Update, consultez : [http://exist-db.org/exist/apps/doc/update_ext.xml](http://exist-db.org/exist/apps/doc/update_ext.xml). |

Scénario associé

Aucun scénario n'est disponible pour la version Standard de ce composant.
**tExternalSortRow**

Ce composant trie les données d'entrée en se basant sur une ou plusieurs colonne(s), par type et ordre de tri, à l'aide d'une application externe de tri.

**Propriétés du tExternalSortRow Standard**

Ces propriétés sont utilisées pour configurer le tExternalSortRow s'exécutant dans le framework de Jobs Standard.

Le composant tExternalSortRow Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

**Basic settings**

| **Schema et Edit Schema** | **Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.**

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **Repository Content**.

Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent. |
| **Built-in** | **Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.** |
| **Repository** | **Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le Guide utilisateur du Studio Talend.** |
| **File Name** | **Chemin d'accès et nom du fichier à traiter, et/ou variable à utiliser.** |
Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Field separator</th>
<th>Caractère, chaîne ou expression régulière séparant les champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>External command “sort” path</td>
<td>Renseignez le chemin d’accès au fichier externe contenant l’algorithme de tri à utiliser.</td>
</tr>
<tr>
<td>Criteria</td>
<td>Cliquez sur [+] pour ajouter autant de lignes que nécessaire pour mettre en place le tri. La première colonne définie dans le schéma est sélectionnée par défaut.</td>
</tr>
<tr>
<td>Schema column</td>
<td>Sélectionnez la colonne de votre schéma sur laquelle vous souhaitez baser votre tri. Notez que l’ordre est important car il détermine la priorité de tri.</td>
</tr>
<tr>
<td>Sort type</td>
<td>Numérique ou Alphabétique.</td>
</tr>
<tr>
<td>Order</td>
<td>Ordre ascendant ou descendant.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| Maximum memory | Renseignez la taille de la mémoire physique que vous voulez allouer au processus de tri. |
| Temporary directory | Définissez l’endroit où stocker les fichiers temporaires. |
| Set temporary input file directory | Cochez la case pour afficher le champ dans lequel vous pouvez spécifier le répertoire du fichier d’entrée temporaire. |
| Add a dummy EOF line | Cochez cette case lorsque vous utilisez le composant tAggregateSortedRow. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Variables d’erreur générées par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</th>
</tr>
</thead>
</table>
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant traite des flux de données, il nécessite donc un composant d’entrée et un composant de sortie. |

**Scénario associé**

Pour un scénario associé, consultez **tSortRow** à la page 3794.
Ce composant génère des colonnes multiples à partir d’une colonne String délimitée

**Propriétés du tExtractDelimitedFields Standard**

Ces propriétés sont utilisées pour configurer le tExtractDelimitedFields s’exécutant dans le framework de Jobs Standard.

Le composant tExtractDelimitedFields Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Field to split</th>
<th>Sélectionnez un champ entrant à partir de la liste Field to split.</th>
</tr>
</thead>
</table>
| Ignore NULL as the source data | Cochez cette case afin d’ignorer les valeurs Null dans les données source.  
Décochez cette case afin de générer les enregistrements s Null correspondant aux valeurs Null dans les données source. |
| Field separator | Définissez le séparateur de champs.  
**Remarque :**  
La syntaxe regex utilise un certain nombre de caractères spéciaux comme opérateurs. Si vous utilisez l’un de ces opérateurs comme séparateur de champs, il est nécessaire de protéger le caractère par un double antislash. Utilisez par exemple "\|" au lieu de "|". |
| Die on error | Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.  
Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject. |
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• **View schema** : sélectionnez cette option afin de voir le schéma.  
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales. |
**Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


---

### Advanced settings

**Advanced separator(for number)**

Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers est une virgule (,) et le séparateur décimal est un point (.)

**Trim columns**

Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.

**Check each row structure against schema**

Cochez cette case pour vérifier si le nombre total de colonnes pour chaque ligne correspond au schéma. S'il ne correspond pas, un message d'erreur s'affichera dans la console.

**Validate date**

Cochez cette case pour vérifier strictement le format de la date par rapport au schéma d'entrée.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.

---

### Global Variables

**Global Variables**

**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace**
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant traite des flux de données, il nécessite donc un composant d’entrée et un composant de sortie. Il permet d’extraire des données d’un champ délimité à l’aide d’un lien Row > Main, et permet de créer un flux de rejet avec un lien Row > Reject, filtrant les données dont le type ne correspond pas au type défini. |

**Scénario : Extraire une colonne String délimitée d’une table d’une base de données**

Ce scénario décrit un Job écrivant des données, y compris une colonne String délimitée dans une table d’une base de données MySQL et affiche les données dans la console, puis extrait la colonne String délimitée en plusieurs colonnes et affiche dans la console les données après extraction.

**Construire le Job**

**Procédure**

1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette : un tFixedFlowInput, un tMysqlOutput, un tMysqlInput, un tExtractDelimitedFields et deux composants tLogRow.
2. Reliez le tFixedFlowInput au tMysqlOutput à l’aide d’un lien Row > Main.
3. Répétez l’opération afin de relier le tMysqlOutput au premier tLogRow, reliez le tMysqlInput au tExtractDelimitedFields et le tExtractDelimitedFields au second tLogRow.
Configurer les composants

Alimenter une table d’une base de données MySQL

Procédure

1. Double-cliquez sur le tFixedFlowInput pour ouvrir sa vue Basic settings.

2. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la fenêtre qui s’ouvre, définissez le schéma, en ajoutant trois colonnes : Id de type Integer, ainsi que Name et DelimitedField de type String.

3. Dans la zone Mode, sélectionnez Use Inline Content(delimited file). Dans le champ Content affiché, saisissez les données à écrire dans la base de données. Les données d’entrée comprennent une colonne String délimitée. Dans cet exemple, les données d’entrée se présentent comme suit :

```
1;Adam;32,Component Team,Developer
2;Bill;28,Component Team,Tester
3;Chris;30,Doc Team,Writer
4;David;35,Doc Team,Leader
5;Eddie;33,QA Team,Tester
```

4. Double-cliquez sur le tMysqlOutput pour ouvrir sa vue Basic settings.
5. Renseignez les champs **Host**, **Port**, **Database**, **Username** et **Password** avec les informations de connexion à la base de données.

6. Renseignez le champ **Table** avec le nom de la table à écrire. Dans cet exemple, saisissez **employee**.

7. Sélectionnez **Drop table if exists and create** dans la liste **Action on table**.

8. Double-cliquez sur le premier **tLogRow** pour ouvrir sa vue **Basic settings**.

Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour une lisibilité optimale des résultats.

**Extrait la colonne String délimitée de la table de la base de données en plusieurs colonnes**

**Procédure**

1. Double-cliquez sur le **tMysqlInput** pour ouvrir sa vue **Basic settings**.
2. Renseignez les champs Host, Port, Database, Username et Password avec les informations de connexion à la base de données MySQL.

3. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la fenêtre qui s’ouvre, définissez le schéma du composant tMysqlInput de la même manière que celui du tMysqlOutput.

4. Dans le champ Table Name, saisissez le nom de la table dans laquelle ont été écrites les données. Dans cet exemple, saisissez employee.

5. Cliquez sur le bouton Guess Query afin de renseigner le champ Query avec l'instruction de requête SQL à exécuter sur la table spécifiée. Dans cet exemple, la requête se présente comme suit:

```
SELECT `employee`.`Id`,
       `employee`.`Name`,
       `employee`.`DelimitedField`
FROM `employee`
```

6. Double-cliquez sur le tExtractDelimitedFields pour ouvrir sa vue Basic settings.
7. Dans la liste **Field to split**, sélectionnez la colonne String délimitée à extraire. Dans cet exemple, sélectionnez **DelimitedField**.
   Dans le champ **Field separator**, saisissez le séparateur utilisé pour séparer les champs, dans la colonne String délimité. Dans cet exemple, saisissez „,‟.
8. Cliquez sur le bouton [ ] à côté du champ **Edit schema** et, dans la fenêtre qui s’ouvre, définissez le schéma en ajoutant cinq colonnes : *Id* de type **Integer**, ainsi que *Name*, *Age*, *Team* et *Title* de type **String**.

Dans cet exemple, la colonne **String** délimitée **DelimitedField** est divisée en trois colonnes *Age*, *Team* et *Title*. Les colonnes *Id* et *Name* sont également conservées.

Cliquez sur **OK** pour fermer l’éditeur de schéma et acceptez la propagation proposée par la boîte de dialogue.

9. Double-cliquez sur le second **tLogRow** pour ouvrir sa vue **Basic settings**.

Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour une lisibilité optimale des résultats.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder le Job.
2. Exécutez le Job en appuyant sur **F6** ou en cliquant sur le bouton **Run** de l’onglet **Run**.
Comme dans la capture d'écran, les données primitives d'entrée et les données après extraction sont affichées dans la console. La colonne **String** délimitée *DelimitedField* est divisée en trois colonnes puis extraite : *Age*, *Team* et *Title*.
tExtractJSONFields

Ce composant extrait les données souhaitées des champs JSON en se basant sur la requête JSONPath ou XPath.

Propriétés du tExtractJSONFields Standard

Ces propriétés sont utilisées pour configurer le tExtractJSONFields s’exécutant dans le framework de Jobs Standard.

Le composant tExtractJSONFields Standard appartient à la famille .

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>: propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>: sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
</tbody>
</table>

Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur Edit schema pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en mode built-in.

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content.
**Read By**

Sélectionnez un moyen d'extraire les données JSON du fichier.

- **JsonPath** : extrait les données JSON en se basant sur la requête JSONPath. Lorsque cette option est sélectionnée, vous devez sélectionner une version de l'API JSONPath dans la liste **API version**. Il est recommandé de lire les données via JSONPath afin d'améliorer les performances.

- **Xpath** : extrait les données JSON en se basant sur la requête XPath.

**JSON field**

Liste des champs JSON à extraire.

**Loop Jsonpath query**

Spécifiez le nœud JSONPath sur lequel se base la boucle.

Si vous avez sélectionné **Xpath** dans la liste déroulante **Read by**, le champ **Loop Xpath query** s'affiche.

**Mapping**

Renseignez cette table pour mapper les colonnes définies dans le schéma aux nœuds JSON correspondants.

- **Column** : les cellules **Column** sont automatiquement renseignées avec le nom des colonnes définies dans le schéma.


  Cette colonne est disponible uniquement lorsque l'option **JsonPath** est sélectionnée dans la liste **Read By**.

- **XPath query** : spécifie le nœud XPath contenant les données souhaitées.

  Cette colonne est disponible uniquement lorsque l'option **Xpath** est sélectionnée dans la liste **Read By**.

- **Get Nodes** : cochez cette case pour extraire les données JSON de tous les nœuds ou cochez la case à côté d'un nœud spécifique pour en extraire les données.

  Cette colonne est disponible uniquement lorsque l'option **Xpath** est sélectionnée dans la liste **Read By**.

- **Is Array** : cochez cette case lorsque le champ JSON à extraire est un tableau et non un objet.

  Cette colonne est disponible uniquement lorsque l'option **Xpath** est sélectionnée dans la liste **Read By**.

**Die on error**

Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient.

Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer...
les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject.

### Advanced settings

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>NB_LINE</th>
<th>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case Die on error est cochée.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant intermédiaire. Il nécessite un composant d’entrée et un composant de sortie.</th>
</tr>
</thead>
</table>

### Scénario : Récupérer les messages d’erreur lors de l’extraction de données de champs JSON

Dans ce scénario, le tWriteJSONField transforme les données d’entrée en des champs JSON, données ensuite extraites par le tExtractJSONFields. Les messages d’erreur générés par les échecs de l’extraction, notamment les champs et erreurs JSON concernés, sont récupérés via un lien Row > Reject.
**Construire le Job**

**Procédure**

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : un `tFixedFlowInput`, un `tWriteJSONField`, un `tExtractJSONFields` et deux `tLogRow`. Renommez les deux `tLogRow`, respectivement `data_extracted` et `reject_info`.
2. Reliez le `tFixedFlowInput` au `tWriteJSONField` à l'aide d'un lien `Row > Main`.
3. Connectez le composant `tWriteJSONField` au `tExtractJSONFields` à l'aide d’un lien `Row > Main`.
4. Reliez le `tExtractJSONFields` au `data_extracted` à l'aide d'un lien `Row > Main`.
5. Reliez le `tExtractJSONFields` au `reject_info` à l’aide d’un lien `Row > Reject`.

**Configurer les composants**

**Configurer le tFixedFlowInput**

**Procédure**

1. Double-cliquez sur le `tFixedFlowInput` pour afficher sa vue `Basic settings`.

2. Cliquez sur `Edit schema` pour ouvrir l’éditeur de schéma.
Cliquez trois fois sur le bouton [+] pour ajouter trois colonnes, respectivement *firstname*, *lastname* et *dept*, de type string.
Cliquez sur OK pour fermer l'éditeur.

3. Sélectionnez Use Inline Content et saisissez les données ci-dessous dans le champ **Content** :

```
Andrew;Wallace;Doc
John;Smith;R&D
Christian;Dior;Sales
```

**Configurer le tWriteJSONField**

**Procédure**

1. Cliquez sur le composant **tWriteJSONField** pour afficher sa vue **Basic settings**.

2. Cliquez sur **Configure JSON Tree** pour ouvrir l'éditeur d'arborescence XML.
Le schéma du `tFixedFlowInput` apparaît dans le panneau **Linker source**.

3. Dans le panneau **Linker target**, cliquez sur le rootTag par défaut et saisissez `staff`, le nœud racine du champ JSON à générer.

4. Cliquez-droit sur `staff` et sélectionnez **Add Sub-element** dans le menu contextuel.

5. Dans la boîte de dialogue qui s’ouvre, saisissez le nom du sous-nœud, *firstname*.

Répétez ces étapes pour ajouter deux autres sous-nœuds, respectivement `lastname` et `dept`.

6. Cliquez-droit sur `firstname` et sélectionnez **Set As Loop Element** dans le menu contextuel.

7. Déposez la ligne `firstname` du panneau **Linker source** dans son homonyme du panneau **Linker target**.

Dans la boîte de dialogue qui s’ouvre, sélectionnez **Add linker to target node**.

Cliquez sur **OK** pour fermer la boîte de dialogue.

8. Répétez les étapes pour relier les deux autres éléments.

Cliquez sur **OK** pour fermer l’éditeur d’arborescence XML.

![Image](image1.png)


**Configurer le tExtractJSONFields**

**Procédure**

1. Double-cliquez sur le *tExtractJSONFields* pour afficher sur la vue *Basic settings*.

![Image](image2.png)

3. Cliquez trois fois sur le bouton [+] dans le panneau droit pour ajouter trois colonnes, respectivement *firstname*, *lastname* et *dept*, qui contiendront les données de leur nœud homonyme dans le champ JSON *staff*.  
Cliquez sur **OK** pour fermer l’éditeur.

4. Dans la boîte de dialogue [Propagate], cliquez sur **Yes** pour propager le schéma aux composants suivants.

5. Dans le champ **Loop XPath query**, saisissez "/staff", le nœud racine des données JSON.

6. Dans la zone **Mapping**, saisissez le nom du nœud des données JSON dans la colonne **XPath query**. Les données de ces nœuds seront extraites et passées à leurs colonnes homonymes définies dans le schéma de sortie.

7. Définissez la requête XPath "*firstname*" pour la colonne *firstname*, "*lastname*" pour la colonne *lastname* et "" pour la colonne *dept*. Notez que "" n’est pas une requête XPath valide et conduit à des erreurs d’exécution.

**Configurer les composants tLogRow**

**Procédure**

1. Double-cliquez sur le composant *data_extracted* pour afficher sa vue **Basic settings**.
2. Sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.
3. Effectuez la même étape pour l’autre composant **tLogRow, reject_info**.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Cliquez sur **F6** pour exécuter le Job.

Le lien de rejet montre des détails, comme les données extraites, les champs JSON dont les données ne sont pas extraites et causent l’échec de l’extraction.

**Scénario 2 : Collecter des données de votre réseau social favori**

Dans ce scénario, le **tFileInputJSON** récupère le nœud **friends** d’un fichier JSON contenant les données d’un utilisateur Facebook et le **tExtractJSONFields** extrait les données du nœud **friends** afin d’obtenir des données plates en sortie.

**Relier les composants**

**Procédure**

1. Déposez les composants suivantes de la **Palette** dans l’espace de modélisation graphique : un **tFileInputJSON**, un **tExtractJSONFields** et un **tLogRow**.
2. Reliez le **tFileInputJSON** au **tExtractJSONFields** à l’aide d’un lien **Row > Main**.
3. Reliez le **tExtractJSONFields** au **tLogRow** à l’aide d’un lien **Row > Main**.
Configurer les composants

Procédure

1. Double-cliquez sur le tFileInputJSON pour afficher sa vue Basic settings.

2. Cliquez sur Edit schema pour ouvrir l'éditeur de schéma.

Cliquez sur le bouton [+] pour ajouter une colonne, que vous nommez friends, de type String.
Cliquez sur OK pour fermer l'éditeur.

3. Cliquez sur le bouton [...] pour parcourir votre système jusqu'au fichier JSON, facebook.json dans cet exemple :

```json
{ 
  "user": { 
    "id": "9999912398",
    "name": "Kelly Clarkson",
    "friends": [
      { 
        "name": "Tom Cruise",
```
4. Décochez la case **Read by XPath**.

Dans la table **Mapping**, saisissez la requête JSONPath `$.user.friends[*]`, dans la colonne **JSONPath query**, pour la ligne **friends**, permettant de récupérer le nœud **friends** complet du fichier source.

5. Double-cliquez sur le **tExtractJSONFields** pour afficher sa vue **Basic settings**.

6. Cliquez sur **Edit schema** pour ouvrir l’éditeur de schéma.

8. Dans la boîte de dialogue [Propagate] qui s’ouvre, cliquez sur Yes pour propager le schéma aux composants suivants.

9. Dans le champ Loop XPath query, saisissez "/likes/data".

10. Dans la zone Mapping, saisissez les requêtes des nœuds JSON dans la colonne XPath query. Les données de ces nœuds seront extraites et passées aux colonnes du même nom, dans le schéma de sortie.

11. Configurez la requête XPath ".//id" (interrogeant le nœud "/friends/id") pour la colonne id.

   Configurez également la requête ".//name" (interrogeant le nœud "friends/name") pour la colonne name, ainsi que les requêtes "id" pour la colonne like_id, "name" pour la colonne like_name et "category" pour la colonne like_category.

12. Double-cliquez sur le tLogRow pour afficher sa vue Basic settings.
13. Dans la zone Mode, sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.
2. Cliquez sur **F6** pour exécuter le Job.

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>like_id</th>
<th>like_name</th>
<th>like_category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>Tom Cruise</td>
<td>1036360939053996</td>
<td>The Shavekank Redemption</td>
<td>Movie</td>
</tr>
<tr>
<td>1234</td>
<td>Tom Cruise</td>
<td>471389562899413</td>
<td>Positive Retribution</td>
<td>Community</td>
</tr>
<tr>
<td>1234</td>
<td>Tom Hanks</td>
<td>1360090231490910</td>
<td>Janelle Wang</td>
<td>Journalist</td>
</tr>
<tr>
<td>1234</td>
<td>Tom Hanks</td>
<td>305948749433410</td>
<td>Now With Alex Wagner</td>
<td>TV show</td>
</tr>
</tbody>
</table>

Comme affiché ci-dessus, les données relatives aux amis de l’utilisateur Facebook Kelly Clarkson sont correctement extraites.
tExtractPositionalFields

Ce composant extrait des données et génère de multiples colonnes à partir d’une chaîne de caractères formatée, à l’aide de champs positionnels.

Le tExtractPositionalFields génère des colonnes multiples à partir d’une colonne donnée en utilisant des champs positionnels.

Propriétés du tExtractPositionalFields Standard

Ces propriétés sont utilisées pour configurer le tExtractPositionalFields s’exécutant dans le framework de Jobs Standard.

Le composant tExtractPositionalFields Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Sélectionnez dans la liste Field le champ entrant à extraire.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore NULL as the source data</td>
<td>Cochez cette case afin d’ignorer les valeurs Null dans les données source. Décochez cette case afin de générer les enregistrement s Null correspondant aux valeurs Null dans les données source.</td>
</tr>
<tr>
<td>Customize</td>
<td>Cochez cette case pour personnaliser le format des données du fichier positionnel et renseignez les colonnes du tableau Formats. <strong>Column</strong> : Sélectionnez la colonne que vous souhaitez personnaliser. <strong>Size</strong> : Saisissez la taille correspondant à la colonne. <strong>Padding char</strong> : saisissez, entre guillemets, le caractère de remplissage à supprimer du champ. Le caractère par défaut est un espace. <strong>Alignment</strong> : Sélectionnez le paramètre d’alignement approprié.</td>
</tr>
<tr>
<td>Pattern</td>
<td>Renseignez le modèle à utiliser pour l’extraction. Un modèle est constitué de valeurs de longueur séparées par des virgules et interprétées comme une chaîne de caractères entre guillemets. Assurez-vous que les valeurs saisies dans ces champs correspondent au schéma défini.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien <strong>Row &gt; Reject</strong>.</td>
</tr>
</tbody>
</table>
### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

---

### Built-In

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository

**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

---

### Advanced settings

<table>
<thead>
<tr>
<th>Advanced separator(for number)</th>
<th>Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers est une virgule (,) et le séparateur décimal est un point (.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim Column</td>
<td>Cochez cette case afin de supprimer les espaces en début et en fin de champ, dans toutes les colonnes.</td>
</tr>
<tr>
<td>Check each row structure against schema</td>
<td>Cochez cette case afin de vérifier que le nombre total de colonnes est cohérent par rapport au schéma. Si le nombre n'est pas cohérent, un message d'erreur s'affiche dans la console.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

---

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est</th>
</tr>
</thead>
</table>
une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.

La variable **NB_LINE** n’est pas disponible en version Map/Reduce.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | Ce composant traite des flux de données, il nécessite donc un composant d’entrée et un composant de sortie. Il permet d’extraire des données d’un champ délimité à l’aide d’un lien **Row > Main**, et permet de créer un flux de rejet avec un lien **Row > Reject**, filtrant les données dont le type ne correspond pas au type défini. |

### Scénario associé

Pour un scénario associé, consultez Scénario : Extraire des noms, des domaines et domaines de premier niveau à partir d’adresses e-mail à la page 1019.
tExtractRegexFields

Ce composant extrait des données et génère des colonnes multiples à partir d’une chaîne de caractères formatée, à l’aide de correspondances regex.

Propriétés du tExtractRegexFields Standard

Ces propriétés sont utilisées pour configurer le tExtractRegexFields s’exécutant dans le framework de Jobs Standard.

Le composant tExtractRegexFields Standard appartient aux familles Data Quality et Processing.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Field to split</th>
<th>Sélectionnez un champ entrant à diviser dans la liste Field to split.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regex</td>
<td>Renseignez l’expression régulière en fonction du langage de programmation que vous utilisez.</td>
</tr>
</tbody>
</table>

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre *Repository Content*.

Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.

**Avertissement** :

Vérifiez que le schéma de sortie ne contient pas de colonne ayant le même nom que la colonne d’entrée à diviser. Sinon, l’expression régulière ne fonctionnera pas comme supposé.
### Advanced settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Die on error</strong></td>
<td>Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien <strong>Row &gt; Reject</strong>.</td>
</tr>
<tr>
<td><strong>Check each row structure against schema</strong></td>
<td>Cochez cette case afin de vérifier que le nombre total de colonnes est cohérent par rapport au schéma. Si le nombre n'est pas cohérent, un message d'erreur s'affiche dans la console.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global Variables</strong></td>
<td></td>
</tr>
</tbody>
</table>

**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Règle d'utilisation</strong></td>
<td>Ce composant traite des flux de données, il nécessite donc un composant d'entrée et un composant de sortie. Il permet d'extraire des données d'un champ délimité à l'aide d'un lien <strong>Row &gt; Main</strong>, et permet de créer un flux de rejet avec un lien <strong>Row &gt; Reject</strong>, filtrant les données dont le type ne correspond pas au type défini.</td>
</tr>
</tbody>
</table>
Scénario : Extraire des noms, des domaines et domaines de premier niveau à partir d'adresses e-mail

Ce scénario décrit un Job à trois composants dans lequel le tExtractRegexFields est utilisé pour définir une expression régulière correspondant à l'une des colonnes des données d'entrée. Dans ce scénario, il s'agit de la colonne email. Le composant tExtractRegexFields est également utilisé pour trouver les correspondances avec ladite expression régulière. Cette expression comprend des identifiants de champ permettant de séparer le nom d'utilisateur, le nom de domaine ainsi que le nom du domaine de premier niveau pour chaque adresse e-mail. Si l'adresse e-mail est valide, ces trois parties sont extraites et affichées sur la console dans trois colonnes différentes. Les données des deux autres colonnes d'entrée, id et age, sont également extraites puis redirigées.

Constructire le Job

Procédure

1. A partir de la Palette, cliquez-déposez les composants suivants dans l'espace graphique : le tFileInputDelimited, le tExtractRegexFields et le tLogRow.
2. Connectez le tFileInputDelimited au tExtractRegexFields à l'aide d'un lien Row > Main, puis reliez le tExtractRegexFields au tLogRow via un autre lien Row > Main.

Configurer les composants

Procédure

1. Dans l'espace graphique, double-cliquez sur le composant tFileInputDelimited pour définir sa configuration de base.

2. Cliquez sur le bouton [...] à côté du champ File Name/Stream pour définir le chemin d'accès au fichier d'entrée.
Dans ce scénario, le fichier d’entrée utilisé est test4. Il s’agit d’un fichier texte comportant trois colonnes : id, email et age.

Pour plus d’informations, consultez tFileInputDelimited à la page 1067.


4. Dans l’espace graphique, double-cliquez sur le composant tExtractRegexFields pour définir sa configuration de base (Basic settings).

5. A partir de la liste déroulante Field to split, sélectionnez la colonne à séparer, à savoir email dans ce scénario.

6. Dans le panneau Regex, renseignez l’expression régulière à utiliser pour effectuer la correspondance. Dans ce scénario, une expression régulière "([a-z]*)@([a-z]*).([a-z]*)" est utilisée pour mettre en correspondance les trois parties d’une adresse e-mail : utilisateur, domaine et domaine de premier niveau.


7. Cliquez sur Edit schema pour ouvrir la boîte de dialogue [Schema of tExtractRegexFields], cliquez cinq fois sur le bouton [+] pour ajouter cinq colonnes au schéma.

Dans ce scénario, vous allez diviser la colonne d’entrée email en trois colonnes dans le flux de sortie, name, domain et tld. Les deux autres colonnes d’entrée seront extraites telles qu’elles sont.
8. Double-cliquez sur le composant tLogRow pour ouvrir sa vue Component.
9. Dans la zone Mode, sélectionnez Table (print values in cells of a table).

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl+S pour enregistrer votre Job.

**Résultats**

Le composant tExtractRegexFields effectue une correspondance entre toutes les adresses e-mail données et l’expression régulière définie, puis extrait le nom d’utilisateur, le domaine et le domaine de premier niveau, qu’il affiche sur la console dans trois colonnes séparées. Les deux autres colonnes, id et age, sont extraites telles quelles.
tExtractXMLField

Ce composant lit les données XML structurées contenues dans un champ XML et envoie les données comme défini dans le schéma au composant suivant.

Propriétés du tExtractXMLField Standard

Ces propriétés sont utilisées pour configurer le tExtractXMLField s’exécutant dans le framework de Jobs Standard.

Le composant tExtractXMLField Standard appartient aux familles Processing et XML.

Le composant de ce framework est toujours disponible.

Basic settings

| Property type | Peut être Built-In ou Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|              | • View schema : sélectionnez cette option afin de voir le schéma.
|              | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
|              | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

Built-In : propriétés utilisées ponctuellement.

Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.

Les champs suivants sont alors pré-remplis à l'aide des données collectées.

Schema type et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.
## Repository
Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

### XML field
Nom du champ XML à traiter.
Voir également le *Guide utilisateur du Studio Talend*

### Loop XPath query
Nœud de l’arborescence XML sur lequel est basé la boucle.

### Mapping
- **Column**: reflète le schéma tel qu’il est défini dans l’éditeur Edit Schema.
- **XPath Query**: Saisissez les champs à extraire de la structure Xml d’entrée.
- **Get nodes**: Cochez cette case pour récupérer le contenu XML de tous les nœuds courants spécifiés dans le champ *Xpath Query*, ou cochez la case correspondant à un nœud spécifique pour ne récupérer que le contenu du nœud sélectionné.

### Limit
Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n’est lue ou traitée.

### Die on error
Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.
Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.
Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.

### Advanced settings

<table>
<thead>
<tr>
<th>Ignore the namespaces</th>
<th>Cochez cette case pour ignorer les espaces de noms lors de la lecture et de l’extraction des données XML.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Global Variables

| Global Variables | **ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  
|-------------------|--------------------------------------------------------------------------------------------------|
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant doit être utilisé en composant intermédiaire. Il nécessite donc un composant d’entrée et de sortie. Il permet d’extraire des données d’un champ délimité à l’aide d’un lien Row > Main, et permet de créer un flux de rejet avec un lien Row > Reject, filtrant les données dont le type ne correspond pas au type défini. |

**Scénario 1 : Extraire les données XML d’un champ d’une table de base de données**

Ce scénario est composé de trois composants et permet de lire la structure XML contenue dans les champs d’une base de données et d’en extraire les données.

![Diagramme de scénario 1](image)

**Procédure**

**Procédure**

1. Glissez les composants tMysqlInput de la famille Database > Mysql, tExtractXMLField de la famille Xml et un composant tFileOutputDelimited de la famille File > Output de la Palette dans le Job designer.

   Reliez les composants entre eux via des connexions de type Main > Row.

2. Double-cliquez sur le composant tMysqlInput pour paramétrer ses propriétés dans l’onglet Basic settings de la vue Component :
3. Si votre schéma est déjà stocké sous le nœud **Db Connections** dans le **Repository**, sélectionnez l’option **Repository** dans le champ **Schema** puis choisissez les métadonnées appropriées à partir de la liste.

Pour plus d’informations concernant le stockage des métadonnées de schéma dans la vue Repository, consultez le **Guide utilisateur du Studio Talend**.

Si vous n’avez encore défini aucun schéma, sélectionnez l’option **Built-in** dans les listes **Schema** et **Property Type** et saisissez manuellement les informations de connexion et la structure des données dans un schéma. Pour plus d’informations sur les propriétés du composant tMysqlInput, consultez tMysqlInput à la page 2631.

4. Dans le champ **Table Name**, saisissez le nom de la table dont les champs contiennent les données XML. Dans ce scénario, la table se nomme **customerdetails** et ne contient qu’une seule colonne.

5. Cliquez sur le bouton **Guess Query** pour récupérer la requête correspondant à votre schéma.

6. Dans le Job designer, double-cliquez sur le composant tExtractXMLField pour paramétrer ses propriétés dans l’onglet **Basic settings**.

7. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton [...] à côté du champ **Edit schema** pour consulter le schéma.

La colonne **Column** du tableau **Mapping** sera automatiquement renseignée avec la description du fichier que vous avez configuré.
8. Dans la liste **Xml field**, sélectionnez la colonne dont vous souhaitez extraire le contenu XML. La colonne se nomme *CustomerDetails* et est l’unique colonne de la table *customerdetails*.

Dans le champ **Loop XPath query**, saisissez le nœud de l’arborescence sur lequel effectuer la boucle afin de récupérer les données.

Dans la colonne **Xpath query**, saisissez entre guillemet le nœud du champ XML contenant les données que vous voulez extraire. Dans ce scénario, l’objectif est d’extraire les noms des clients, saisissez donc “*CustomerName*” qui correspond à l’élément XML contenant cette information.

9. Dans le Job designer, double-cliquez sur le composant **tFileOutputDelimited** pour paramétrer ses propriétés dans l’onglet **Basic settings**.

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Built-In</th>
<th><img src="icon.png" alt="icon" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Output Stream</td>
<td><img src="checkbox.png" alt="checkbox" /></td>
<td></td>
</tr>
<tr>
<td>File Name</td>
<td><img src="input.png" alt="input" /></td>
<td>“D:\04_Jobs\CustomerNames.csv”</td>
</tr>
<tr>
<td>Row Separator</td>
<td>“,”</td>
<td>Field Separator</td>
</tr>
<tr>
<td>Append</td>
<td><img src="checkbox.png" alt="checkbox" /></td>
<td>Include Header</td>
</tr>
<tr>
<td>Compress as zip File</td>
<td><img src="checkbox.png" alt="checkbox" /></td>
<td><img src="icon.png" alt=" Sync columns" /></td>
</tr>
</tbody>
</table>

10. Dans le champ **File Name**, indiquez le répertoire et le nom du fichier de sortie. Pour ce scénario, appelez le fichier *CustomerNames.csv*.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton […] à côté du champ **Edit schema** pour consulter le schéma.

11. Enregistrez le Job et appuyez sur **F6** pour l’exécuter.

**Résultats**

| 1 | Griffith Paving and Sealcoating |
| 2 | Bill’s Dive Shop               |
| 3 | Childress Child Day Care       |
| 4 | Facelift Kitchen and Bath      |
| 5 | Terrini & Son Auto and Truck   |
| 6 | Kermit the Pet Shop            |
| 7 | Tub’s Furniture Store          |
| 8 | Togule & Myerson Ltd           |
| 9 | Childress Child Day Care       |
| 10| Elle Hypnosis and Therapy Cent |
| 11| Lennox Air Pollution Control   |

Le Job retourne les noms des clients contenus sous le nœud *CustomerName* du champ *CustomerDetails* de la table.

**Scénario 2 : Extraire les données valides et les données erronées à partir d’un champ XML dans fichier délimité**

Ce scénario est composé de quatre composants et permet de lire la structure XML contenue dans les champs d’un fichier délimité, d’en extraire les données et de rejeter les données erronées.
**Procédure**

1. A partir de la Palette, glissez dans le Job designer les composants tFileInputDelimited de la famille File > Input, tExtractXMLField de la famille Xml, tFileOutputDelimited de la famille File > Output et tLogRow de la famille Logs & Errors.

   Connectez les trois premiers composants à l’aide de liens Row > Main

   Reliez le tExtractXMLField au tLogRow à l’aide d’un lien Row > Reject.

2. Double-cliquez sur le composant tFileInputDelimited pour paramétrer ses propriétés dans l’onglet Basic settings.

   Sélectionnez l’option Built-in à partir de la liste déroulante Schema et renseignez manuellement les informations concernant le fichier délimité dans les champs correspondants.

   Cliquez sur le bouton [...] à côté du champ Edit schema pour afficher la boîte de dialogue permettant de définir la structure des données, comme présenté ci-dessus.

   Cliquez sur le bouton [+] pour ajouter autant de lignes que nécessaire à la structure des données.

   Dans cet exemple, il n’y a qu’une seule colonne dans le schéma : xmlStr.

   Cliquez sur OK pour valider vos changements et fermer la boîte de dialogue.
Remarque :
Si vous avez préalablement stocké votre schéma dans le répertoire Metadata sous File delimited, sélectionnez l’option Repository dans la liste déroulante Schema puis cliquez sur le bouton […] à côté du champ pour afficher la boîte de dialogue [Repository Content] à partir de laquelle vous pouvez sélectionner un schéma pertinent. Cliquez sur OK pour fermer la boîte de dialogue et remplir les champs automatiquement avec les métadonnées contenues dans le schéma.

Pour plus d’informations sur le stockage des schémas de métadonnées dans le Repository, consultez le Guide utilisateur du Studio Talend.

4. Dans le champ File Name, renseignez le chemin d’accès ou cliquez sur le bouton [...] pour parcourir vos dossiers jusqu’au fichier à traiter. Dans cet exemple, il s’agit du fichier CustomerDetails_Error. Ce fichier csv stocké localement contient une suite de lignes simples au format XML, séparées par un double retour chariot.

Dans le champ Row Separator, renseignez le séparateur de lignes, ici le double retour chariot. Dans le champ Field Separator, renseignez le séparateur de champ, aucun ici.

Dans cet exemple, il n’y a pas d’en-tête (Header), de pied de page (Footer) ou de nombre limite de lignes à traiter (Limit).

5. Dans l’espace graphique, double-cliquez sur le composant tExtractXMLField pour paramétrer ses propriétés dans l’onglet Basic settings.

6. Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton [...] à côté du champ Edit schema pour consulter le schéma ou le modifier.

La colonne Column du tableau Mapping sera automatiquement renseignée avec le schéma défini.

7. Dans la liste Xml field, sélectionnez la colonne dont vous souhaitez extraire le contenu XML. Dans cet exemple, la colonne se nomme xmlStr.

Dans le champ Loop XPath query, saisissez le nœud de l’arborescence sur lequel effectuer la boucle afin de récupérer les données.

8. Dans le Job designer, double-cliquez sur le composant tFileOutputDelimited pour paramétrer ses propriétés dans l’onglet Basic settings.
9. Dans le champ **File Name**, indiquez le répertoire et le nom du fichier de sortie. Pour ce scénario, appelez le fichier *CustomerNames_right.csv*.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton [...] à côté du champ **Edit schema** pour consulter le schéma ou le modifier.

10. Dans le Job designer, double-cliquez sur le composant **tLogRow** pour paramétrer ses propriétés dans l’onglet **Basic settings** de la vue **Component**.

11. Cliquez sur le bouton **Sync Columns** pour récupérer le schéma du composant précédent. Pour plus d’informations sur le composant **tLogRow**, consultez **tLogRow** à la page 2105.

12. Enregistrez le Job et appuyez sur **F6** pour l’exécuter.

**Résultats**


[statistics] connecting to socket on port 3845
[statistics] connected

<CustomerDetails>
  <CustomerName>Childress Child Day Care</CustomerName>
</CustomerDetails>

| Error on line 3 of document : The element type "CustomerName" must be terminated by the matching end-tag "</CustomerName>". Nested exception: The element type "CustomerName" must be terminated by the matching end-tag "</CustomerName>". – Line: 8
[statistics] disconnected

Job A ended at 14:22 05/11/2009: [exit code=0]

Le composant **tExtractXMLField** lit et extrait dans un fichier délimité en sortie, *CustomerNames_right*, dans cet exemple, les informations concernant les clients dont la structure XML est correcte, et affiche les lignes erronées dans la console de log.
Ce composant crée une nouvelle archive .zip, .gzip ou .tar.gz à partir d’un ou plusieurs fichiers ou dossiers.

Cette archive peut être compressée en utilisant différentes méthodes de compression.

**Propriétés du tFileArchive Standard**

Ces propriétés sont utilisées pour configurer le tFileArchive s’exécutant dans le framework de Jobs Standard.

Le composant tFileArchive Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory</td>
<td>Saisissez le chemin d’accès au répertoire contenant les fichiers à zipper.</td>
</tr>
<tr>
<td></td>
<td>Ce champ est accessible lorsque le format zip ou tar.gz est sélectionné dans la liste Archive format.</td>
</tr>
<tr>
<td>Subdirectories</td>
<td>Cochez cette case si vous souhaitez ajouter les fichiers des sous-répertoires dans l’archive.</td>
</tr>
<tr>
<td></td>
<td>Ce champ est uniquement accessible lorsque zip est sélectionné dans la liste Archive format.</td>
</tr>
<tr>
<td>Source File</td>
<td>Indiquez le chemin du fichier que vous souhaitez ajouter à l’archive.</td>
</tr>
<tr>
<td></td>
<td>Ce champ est uniquement accessible lorsque gzip est sélectionné dans la liste Archive format.</td>
</tr>
<tr>
<td>Archive file</td>
<td>Indiquez l’emplacement où vous souhaitez créer votre archive.</td>
</tr>
<tr>
<td>Create directory if does not exist</td>
<td>Cochez cette case pour créer le fichier de destination s’il n’existe pas.</td>
</tr>
<tr>
<td>Archive format</td>
<td>Dans la liste, sélectionnez un format d’archive zip, gzip, ou tar.gz.</td>
</tr>
<tr>
<td>Compress level</td>
<td>Choisissez le niveau de compression que vous souhaitez appliquer.</td>
</tr>
<tr>
<td></td>
<td>• Best : la qualité de compression sera optimale mais le temps de compression sera long.</td>
</tr>
<tr>
<td></td>
<td>• Normal : qualité et temps de compression intermédiaires.</td>
</tr>
<tr>
<td></td>
<td>• Fast (no compression) : la compression sera rapide mais la qualité de compression sera amoindrie.</td>
</tr>
<tr>
<td>All files</td>
<td>Cochez cette case si tous les fichiers du répertoire spécifié doivent être ajoutés à l’archive.</td>
</tr>
</tbody>
</table>
Décochez-la pour spécifier dans le tableau **Files** le ou les fichier(s) que vous désirez ajouter à l'archive.

**Filemask** : saisissez le nom de fichier ou le masque de fichier utilisant un caractère spécial ou une expression régulière.

Cette case est accessible lorsque **zip** ou **tar.gz** est sélectionné dans la liste **Archive format**.

### Encoding

 Sélectionnez un encodage à partir de la liste ou sélectionnez **CUSTOM** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

Cette liste est accessible lorsque **zip** est sélectionné dans la liste **Archive format**.

### Overwrite Existing Archive

Cette case est cochée par défaut. Elle vous permet de sauvegarder une archive en remplaçant l’archive existante. Cependant, si vous décochez la case, une erreur sera rapportée, le remplacement d’archive échouera et la nouvelle archive ne pourra être sauvegardée.

Remarque :
Lorsque le remplacement échoue, le Job poursuit son exécution.

### Encrypt files

Cochez cette case si vous souhaitez que le fichier d’archive soit protégée par un mot de passe.

**Encrypt method** : sélectionnez une méthode de cryptage dans la liste **Java Encrypt, Zip4j AES**, ou **Zip4j STANDARD**.

**AES Key Strength** : sélectionnez une taille de clé pour la méthode **Zip4j AES, AES 128** ou **AES 256**.

**Enter Password** : saisissez le mot de passe de cryptage.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Cette case est disponible uniquement lorsque l’option **zip** est sélectionnée dans la liste **Archive format**. Lorsque cette case est cochée, le fichier archive compressé peut être décompressé uniquement par le composant **tFileUnarchive** et non par votre programme habituel. Pour plus d’informations concernant le **tFileUnarchive**, consultez **tFileUnarchive** à la page 1231.

### ZIP64 mode

Cette option permet de créer des archives avec l’extension .zip64. Trois modes sont disponibles :

- **ASNEEDED** : les archives ayant une extension .zip64 sont automatiquement créées selon la taille du fichier.
- **ALWAYS** : les archives ayant l’extension .zip64 sont créées, quelle que soit la taille du fichier.
- **NEVER** : les archives ayant l'extension .zip64 ne sont pas créées, quelle que soit leur taille.

Notez que si la taille du fichier ou la taille totale de l'archive dépasse les 4Go, ou si elle contient plus de 65536 fichiers, vous devez sélectionner le mode **ALWAYS**.

### Advanced settings

| **Use sync flush** | Cochez cette case afin de vider le compresseur avant de vider le flux de sortie. Décrochez cette case pour vider uniquement le flux de sortie.

Cette case est disponible uniquement lorsque l'option `gzip` ou `tar.gz` est sélectionnée dans la liste **Archive format**. |

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| **Global Variables** | **ARCHIVE_FILEPATH** : chemin vers le fichier d'archive. Cette variable est une variable **After** et retourne une chaîne de caractères.

**ARCHIVE_FILENAME** : le nom du fichier archive. Cette variable est une variable **After** et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| **Règle d'utilisation** | Ce composant doit être utilisé en standalone. |

| **Connections** | Liens de sortie (de ce composant à un autre) :

**Row** : Main, Reject, Iterate.

Scénario : Zipper des fichiers à l'aide d'un tFileArchive

Ce scénario met en scène un Job formé d'un composant unique. Il a pour but de zipper des fichiers et de les enregistrer dans le répertoire choisi.

Procédure

1. Cliquez-déposez le composant tFileArchive de la Palette dans l'espace de modélisation graphique.
2. Double-cliquez sur le composant afin d'afficher sa vue Component.

3. Dans le champ Directory, cliquez sur [...], parcourez votre répertoire et sélectionnez le dossier ou le fichier que vous souhaitez compresser.
4. Cochez la case Subdirectories si vous désirez inclure dans l’archive les fichiers des sous-dossiers.
5. Renseignez ensuite le champ Archive file, en indiquant le répertoire de destination et le nom de votre archive.
6. Cochez la case Create directory if not exists si votre répertoire de destination n’existe pas encore et que vous souhaitez le créer.
7. Dans la liste **Compress level**, sélectionnez le niveau de compression que vous désirez appliquer à votre archive. Dans cet exemple, utilisez le niveau **normal**.

8. Décochez la case **All Files** si vous ne souhaitez compresser que des fichiers spécifiques.

9. Ajoutez une ligne dans le tableau à l'aide du bouton [+], et cliquez sur le nom qui s'affiche. Saisissez entre étoiles une partie du nom du ou des fichier(s) que vous souhaitez compresser.

10. Appuyez sur **F6** afin d'exécuter votre Job.

**Résultats**

Le **tArchiveFile** a compressé les fichiers sélectionnés et a créé le dossier dans le répertoire indiqué.
tFileCompare

Ce composant compare deux fichiers et fournit des informations de comparaison en fonction d’un schéma en lecture seule.

**Propriétés du tFileCompare Standard**

Ces propriétés sont utilisées pour configurer le tFileCompare s’exécutant dans le framework de Jobs Standard.

Le composant tFileCompare Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule.</th>
</tr>
</thead>
<tbody>
<tr>
<td>File to compare</td>
<td>Le chemin d’accès au fichier à vérifier.</td>
</tr>
<tr>
<td>Reference file</td>
<td>Le chemin d’accès du fichier sur lequel la comparaison est basé.</td>
</tr>
<tr>
<td>If differences are detected, display</td>
<td>If no difference detected, display</td>
</tr>
<tr>
<td>Print to console</td>
<td>Cochez cette case pour afficher le message saisi.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| Global Variables | DIFFERENCE : résultat de la comparaison. Cette variable est une variable Flow et retourne un booléen.  
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

#### Règle d’utilisation

Ce composant peut être utilisé en standalone mais il est généralement relié à un composant de sortie pour regrouper les informations de comparaison.

#### Connections

Liens de sortie (de ce composant à un autre) :
- **Row** : Main.

Liens d’entrée (d’un autre composant à celui-ci) :
- **Row** : Main, Reject, Iterate.

Pour plus d’informations concernant les liens, consultez la section relative aux types de connexions, dans le **Guide utilisateur du Studio Talend**.

### Scénario : Comparer des fichiers dézippés

Ce scénario décrit un Job permettant de dézipper un fichier et de le comparer avec un fichier de référence pour s’assurer qu’il n’a pas été modifié. La sortie de cette comparaison est stockée dans un fichier délimité et un message est affiché dans la console.

#### Procédure

1. Cliquez et déposez les composants suivants : **tFileUnarchive**, **tFileCompare** et **tFileOutputDelimited**.
2. Reliez les composants **tFileUnarchive** et **tFileCompare** à l’aide d’une connexion de type **Iterate**.
3. Connectez le composant **tFileCompare** au composant de sortie à l’aide d’un lien **Main row**.
4. Dans l’onglet **Basic settings** du composant tFileUnarchive, renseignez le chemin d’accès au fichier à dézipper.

5. Dans le champ **Extraction Directory**, renseignez le chemin de destination du fichier dézippé.

6. Dans l’onglet Basic settings du composant tFileCompare, paramétrez les propriétés du fichier à comparer. Dans le champ **File to compare**, appuyez sur Ctrl+Espace pour afficher la liste des variables globales. Sélectionnez \$_globals[tFileUnarchive_1]{CURRENT_FILEPATH} ou "((String)globalMap.get("tFileUnarchive_1_CURRENT_FILEPATH"))" en fonction du langage avec lequel vous travaillez, pour récupérer le chemin d’accès du fichier à partir du composant tFileUnarchive.

7. Puis dans le champ **Reference file**, paramétrez le fichier de référence sur lequel est basée la comparaison.

8. Dans les champs **If differences detected, display** et **If no differences detected, display**, paramétrez les messages à afficher lorsque les fichiers sont différents et lorsqu’ils sont identiques, par exemple : "[Job " + JobName + "] Files differ"

9. Cochez la case **Print to console**, pour que le message défini apparaîsse à la fin de l’exécution du Job.

10. Le schéma est en lecture seule et contient les informations standard. Cliquez sur **Edit schema** pour le consulter.

11. Paramétrez le composant de sortie en utilisant des points-virgules comme séparateurs de données.

12. Enregistrez le Job et appuyez sur **F6** pour l’exécuter.
Le message paramétré apparaît dans la console et la sortie contient les informations telles qu'elles ont été définies dans le schéma.

Starting job CompareFiles at 14:11 19/06/2007
[job CompareFiles] Files differ
Job CompareFiles ended at 14:11 19/06/2007 [exit code=0]
Ce composant fait une copie d’un fichier ou d’un dossier source dans un répertoire cible.

**Propriétés du tFileCopy Standard**

Ces propriétés sont utilisées pour configurer le tFileCopy s’exécutant dans le framework de Jobs Standard.

Le composant tFileCopy Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File Name</strong></td>
<td>Spécifiez le chemin d’accès au fichier à copier.</td>
</tr>
<tr>
<td></td>
<td>Ce champ n’apparaît pas lorsque la case <strong>Copy a directory</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Copy a directory</strong></td>
<td>Cochez cette case pour copier un répertoire ainsi que tous les sous-répertoires et fichiers qu’il contient.</td>
</tr>
<tr>
<td><strong>Source directory</strong></td>
<td>Spécifiez le répertoire source à copier.</td>
</tr>
<tr>
<td></td>
<td>Ce champ apparaît uniquement lorsque la case <strong>Copy a directory</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Destination directory</strong></td>
<td>Spécifiez le répertoire dans lequel le fichier source ou le répertoire doit être copié.</td>
</tr>
<tr>
<td><strong>Rename</strong></td>
<td>Cochez cette case pour renommer le fichier copié à la destination.</td>
</tr>
<tr>
<td></td>
<td>Ce champ n’apparaît pas lorsque la case <strong>Copy a directory</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Destination filename</strong></td>
<td>Spécifiez un nouveau nom pour le fichier à copier.</td>
</tr>
<tr>
<td></td>
<td>Ce champ apparaît uniquement lorsque la case <strong>Rename</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Remove source file</strong></td>
<td>Cochez cette case pour supprimer le fichier source après sa copie dans le répertoire de destination.</td>
</tr>
<tr>
<td></td>
<td>Ce champ n’apparaît pas lorsque la case <strong>Copy a directory</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Replace existing file</strong></td>
<td>Cochez cette case pour écraser le fichier déjà existant.</td>
</tr>
<tr>
<td></td>
<td>Ce champ n’apparaît pas lorsque la case <strong>Copy a directory</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Create a directory if it doesn’t exist</strong></td>
<td>Cochez cette case pour créer le répertoire de destination spécifié s’il n’existe pas déjà.</td>
</tr>
<tr>
<td></td>
<td>Ce champ n’apparaît pas lorsque la case <strong>Copy a directory</strong> est cochée.</td>
</tr>
</tbody>
</table>
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Global Variables

| Global Variables | **DESTINATION_FILENAME** : nom du fichier cible. Cette variable est une variable *After* et retourne une chaîne de caractères. |
| **DESTINATION_FILEPATH** : chemin d’accès au fichier cible. Cette variable est une variable *After* et retourne une chaîne de caractères. |
| **SOURCE_DIRECTORY** : répertoire source. Cette variable est une variable *After* et retourne une chaîne de caractères. |
| **DESTINATION_DIRECTORY** : répertoire cible. Cette variable est une variable *After* et retourne une chaîne de caractères. |
| **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. |

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en standalone. |
| Connections | Liens de sortie (de ce composant à un autre) :

| **Row** : Main. |

Liens d’entrée (d’un autre composant à celui-ci) :

| **Row** : Main, Reject, Iterate. |
Scénario : Récupérer un fichier de la corbeille

Ce scénario décrit un Job faisant une boucle sur une liste de fichiers, copie chaque fichier du répertoire source vers le répertoire cible et supprime les fichiers copiés du répertoire source.

Procédure

1. Créez un nouveau Job et ajoutez les composants tFileList et tFileCopy en saisissant leurs noms dans l’espace de modélisation graphique ou en les déposant de la Palette.
2. Reliez tFileList à tFileCopy à l’aide d’une connexion Row > Iterate.
3. Double-cliquez sur le composant tFileList pour ouvrir sa vue Basic settings.

4. Dans le champ Directory, parcourez votre système ou saisissez le répertoire sur lequel sera effectuée la boucle.
5. Double-cliquez sur le composant tFileCopy pour ouvrir sa vue Basic settings.
6. Dans le champ **File Name**, appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales et sélectionnez la variable `tFileList_1.CURRENT_FILEPATH` dans la liste pour remplir le champ avec

\[(String)globalMap.get("tFileList_1_CURRENT_FILEPATH")\].

7. Dans le champ **Destination directory**, parcourrez votre système ou saisissez le répertoire dans lequel doit être copié chaque fichier.

8. Cochez la case **Remove source file** pour supprimer les fichiers ayant été copiés.

9. Appuyez sur **Ctrl+S** pour enregistrer le Job et appuyez sur **F6** pour l’exécuter.

   Tous les fichiers dans le répertoire source spécifié sont copiés dans le répertoire de destination et supprimés du répertoire source.
tFileDelete

Ce composant supprime des fichiers d'un répertoire spécifié.

Propriétés du tFileDelete Standard

Ces propriétés sont utilisées pour configurer le tFileDelete s'exécutant dans le framework de Jobs Standard.

Le composant tFileDelete Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

| **File Name** | Chemin d'accès au fichier à supprimer. Ce champ n'apparaît pas si la case Delete folder ou la case Delete file or folder est cochée. |
| **Directory** | Chemin d'accès au dossier à supprimer. Ce champ est disponible uniquement lorsque la case Delete folder est cochée. |
| **File or directory to delete** | Saisissez le chemin d'accès au fichier ou au dossier que vous souhaitez supprimer. Ce champ est disponible uniquement lorsque la case Delete file or folder est cochée. |
| **Fail on error** | Cochez cette case pour empêcher l'exécution du Job si une erreur survient, par exemple si le fichier à supprimer n'existe pas. |
| **Delete Folder** | Cochez cette case pour afficher le champ Directory, dans lequel indiquer le chemin d'accès au dossier à supprimer. |
| **Delete file or folder** | Cochez cette case pour afficher le champ File or directory to delete, dans lequel indiquer le chemin d'accès au dossier ou au fichier que vous souhaitez supprimer. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables

| **Global Variables** | **DELETE_PATH** : chemin du dossier ou fichier supprimé. Cette variable est une variable After et retourne une chaîne de caractères. |
**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en standalone.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connections</td>
<td>Liens de sortie (de ce composant à un autre) :</td>
</tr>
<tr>
<td></td>
<td><strong>Row</strong> : Main.</td>
</tr>
<tr>
<td></td>
<td>Liens d’entrée (d’un autre composant à celui-ci) :</td>
</tr>
<tr>
<td></td>
<td><strong>Row</strong> : Main, Reject, Iterate.</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant les liens, consultez la section relative aux types de connexion, dans le Guide utilisateur du Studio Talend.

### Scénario : Supprimer des fichiers

Ce scénario décrit un Job très simple supprimant des fichiers d’un répertoire donné.

![Diagramme de flux du scénario](image)

**Procédure**

1. Cliquez et déposez les composants suivants : tFileList, tFileDelete, tJava.
2. Dans le champ **Directory** de l'onglet **Basic settings** du composant **tFileList**, spécifiez le répertoire sur lequel la boucle doit être effectuée.

3. Dans le champ **Filemask**, paramétrez le masque de fichier en "*.txt". Ne cochez aucune case.

4. Dans l'onglet **Basic settings** du composant **tFileDelete**, paramétrez le champ **File Name** afin que le fichier sélectionné par le composant **tFileList** soit supprimé. Ceci permet de supprimer tous les fichiers d'un répertoire spécifié.

5. Appuyez sur Ctrl+Espace pour accéder à la liste des variables globales. La variable adéquate permettant de récupérer le fichier en cours est : 

```
((String)globalMap.get("tFileList_1_CURRENT_FILEPATH"))
```

6. Puis dans le composant **tJava**, définissez le message à afficher dans la sortie standard (la console Run). Dans cet exemple, saisissez le script suivant dans le champ Code :

```
System.out.println( ((String)globalMap.get("tFileList_1_CURRENT_FILE")) + " has been deleted! ");
```

7. Appuyez sur Ctrl+S pour sauvegarder votre Job.

8. Appuyez sur F6 ou sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

```
Starting job FileDelete at 18:29 20/06/2007
16.txt has been deleted!
15.txt has been deleted!
14.txt has been deleted!
13.txt has been deleted!
12.txt has been deleted!
11.txt has been deleted!
10.txt has been deleted!
09.txt has been deleted!
08.txt has been deleted!
07.txt has been deleted!
06.txt has been deleted!
05.txt has been deleted!
04.txt has been deleted!
03.txt has been deleted!
02.txt has been deleted!
01.txt has been deleted!
Job FileDelete ended at 18:29 20/06/2007. [exit code=0]
```

**Résultats**

Le message défini dans le composant **tJava** apparaît dans la console pour chaque fichier supprimé à l'aide du composant **tFileDelete**.
**tFileExist**

Ce composant vérifie si un fichier existe ou non.

**Propriétés du tFileExist Standard**

Ces propriétés sont utilisées pour configurer le tFileExist s'exécutant dans le framework de Jobs Standard.

Le composant tFileExist Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

| File Name       | Chemin d’accès au fichier dont vous voulez vérifier l’existence. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | EXIST : résultat spécifiant si un fichier existe ou non. Cette variable est une variable Flow et retourne un booléen. |

|                  | FILENAME : nom du fichier traité. Cette variable est une variable After et retourne une chaîne de caractères. |

|                  | ERROR MESSAGE : message d’erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |

|                  | Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |

|                  | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |

|                  | Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| Règle d’utilisation | Ce composant peut être utilisé en standalone. |
### Scénario : Vérifier l'existence d'un fichier et le créer s'il n'existe pas

Ce scénario décrit un Job simple qui permet de vérifier si le fichier spécifié existe, d’afficher un message à l’écran si le fichier n’existe pas, de lire les données d’un autre fichier d’entrée spécifié et de les écrire dans un fichier délimité en sortie.

Une boîte de dialogue apparaît pour confirmer que le fichier n’existe pas.

Cliquez sur **OK** pour fermer cette boîte de dialogue et continuer l’exécution du Job. Le fichier manquant, “file1” dans cet exemple, est écrit sous la forme d’un fichier délimité à l’endroit préalablement déterminé.

### Déposer et relier les composants

**Procédure**

1. A partir de la **Palette**, cliquez-déposez les composants suivants dans l’éditeur graphique : le **tFileExist**, le **tFileInputDelimited**, le **tFileOutputDelimited** et le **tMsgBox**.
2. Reliez d’abord le **tFileExist** au **tFileInputDelimited** à l’aide d’un lien de type **OnSubjobOk**, puis au **tMsgBox** à l’aide d’un lien de type **Run If**.
3. Reliez le **tFileInputDelimited** au **tFileOutputDelimited** à l’aide d’un lien de type **Row Main**.

---

**Connections**

<table>
<thead>
<tr>
<th>Liens de sortie (de ce composant à un autre) :</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row</strong> : Main.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Liens d’entrée (d’un autre composant à celui-ci) :</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row</strong> : Main, Reject, Iterate.</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant les liens, consultez la section relatives aux types de connexions, dans le *Guide utilisateur du Studio Talend*. 

---

![Diagramme de connexion](image_url)
Configurer les composants

Procédure

1. Dans l'éditeur graphique, sélectionnez le tFileExist puis cliquez sur la vue Component pour en définir la configuration de base (Basic settings).

2. Dans le champ File name, renseignez le chemin d'accès au fichier ou parcourrez vos dossiers à la recherche du fichier dont vous voulez vérifier l'existence.

3. Dans l'éditeur graphique, sélectionnez le composant tFileInputDelimited puis cliquez sur la vue Component pour en définir la configuration de base (Basic settings).

4. Renseignez le champ File Name en parcourant vos dossiers jusqu'au fichier d'entrée que vous voulez lire.

   Avertissement :

   Si le chemin d'accès du fichier contient des caractères accentués, vous obtiendrez un message d'erreur lors de l'exécution du Job.

5. Définissez les séparateurs de lignes et de champs dans leurs champs respectifs Row separator et Field Separator.

6. Définissez le nombre de lignes d'en-tête (Header), de pied de page (Footer) et le nombre maximal de lignes à traiter (Limit) en fonction de vos besoins. Dans ce scénario, notre tableau comporte une ligne d'en-tête.


   Pour plus d'informations sur les types de schémas, consultez le Guide utilisateur du Studio Talend.
Le schéma du fichier “file2” est composé de cinq colonnes : Num, Ref, Price, Quant et Tax.

8. Dans l’éditeur graphique, sélectionnez le composant tFileOutputDelimited.
9. Cliquez sur la vue Component pour définir la configuration de base (Basic settings) du tFileOutputDelimited.

10. Définissez le champ Property Type en mode Built-in.
11. Dans le champ File name, appuyez simultanément sur Ctrl+Espace pour accéder à la liste des variables et sélectionner FILENAME.
12. Définissez les séparateurs de lignes et de champs dans leurs champs respectifs Row separator et Field Separator.
13. Cochez la case Include Header car, dans cet exemple, le fichier “file2” comprend un en-tête.
14. Définissez le champ Schema en mode Built-in puis cliquez sur le bouton Sync columns pour synchroniser le schéma du fichier de sortie (file1) avec le schéma du fichier d’entrée (file2).
Dans l’éditeur graphique, sélectionnez le composant tMsgBox.

Cliquez sur la vue Component pour définir la configuration de base (Basic settings) du tMsgBox.

Cliquez sur le lien If pour en afficher les propriétés dans la vue Basic settings.

Dans le panneau Condition, appuyez simultanément sur Ctrl+Espace pour accéder à la liste de variable et sélectionner EXISTS. Entrez un point d’exclamation avant la variable pour en inverser le sens.

Enregistrer et exécuter le Job

Procédure
1. Appuyez sur Ctrl+S pour enregistrer votre Job.
2. Appuyez sur F6 ou cliquez sur le bouton Run de l’onglet Run pour l’exécuter.
File does not exist!
tFileFetch

Ce composant récupère un fichier via un protocole donné (HTTP, HTTPS, FTP ou SMB).

**Propriétés du tFileFetch Standard**

Ces propriétés sont utilisées pour configurer le tFileFetch s'exécutant dans le framework de Jobs Standard.

Le composant tFileFetch Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URI</strong></td>
<td>Saisissez l'URI du site sur lequel le fichier doit être récupéré.</td>
</tr>
<tr>
<td><strong>Use cache to save the resource</strong></td>
<td>Cochez cette case pour enregistrer les données dans le cache. Cette option vous permet de traiter directement le flux de données du fichier (en mode stream) sans sauvegarder ce fichier sur votre disque. Elle permet donc un gain de temps et de performance.</td>
</tr>
<tr>
<td><strong>Domain</strong></td>
<td>Saisissez le nom du domaine du serveur Microsoft. Disponible pour le protocole smb.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez l’identifiant utilisateur et le mot de passe de connexion au serveur. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. Disponible pour le protocole smb.</td>
</tr>
<tr>
<td><strong>Destination directory</strong></td>
<td>Sélectionnez le répertoire de destination dans lequel le fichier récupéré sera placé.</td>
</tr>
<tr>
<td><strong>Destination Filename</strong></td>
<td>Saisissez le nouveau nom du fichier récupéré.</td>
</tr>
<tr>
<td><strong>Create full path according to URI</strong></td>
<td>Cette case permet de reproduire l’arborescence de répertoires de l’URI. Pour enregistrer le fichier à la</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>racine de votre</td>
<td>Cochez cette case si vous souhaitez ajouter un ou plusiers en-tête(s) comme conditions de récupération. Dans le tableau <strong>Headers</strong>, saisissez le(s) paramètre(s) d'en-tête HTTP dans le champ <strong>Name</strong> et la valeur correspondante dans le champ <strong>Value</strong>.</td>
</tr>
<tr>
<td>repértoire de</td>
<td>Cochez cette case si vous souhaitez ajouter un ou plusiers en-tête(s) comme conditions de récupération. Dans le tableau <strong>Headers</strong>, saisissez le(s) paramètre(s) d'en-tête HTTP dans le champ <strong>Name</strong> et la valeur correspondante dans le champ <strong>Value</strong>.</td>
</tr>
<tr>
<td>destination,</td>
<td>Cochez cette case si vous souhaitez ajouter un ou plusiers en-tête(s) comme conditions de récupération. Dans le tableau <strong>Headers</strong>, saisissez le(s) paramètre(s) d'en-tête HTTP dans le champ <strong>Name</strong> et la valeur correspondante dans le champ <strong>Value</strong>.</td>
</tr>
<tr>
<td>Décochez cette case</td>
<td>Décochez cette case pour passer les lignes en erreur et terminer le traitement des lignes sans erreur.</td>
</tr>
<tr>
<td>case.</td>
<td>Décochez cette case pour passer les lignes en erreur et terminer le traitement des lignes sans erreur.</td>
</tr>
<tr>
<td><strong>Add header</strong></td>
<td>Cochez cette case si vous souhaitez ajouter un ou plusiers en-tête(s) comme conditions de récupération. Dans le tableau <strong>Headers</strong>, saisissez le(s) paramètre(s) d'en-tête HTTP dans le champ <strong>Name</strong> et la valeur correspondante dans le champ <strong>Value</strong>.</td>
</tr>
<tr>
<td><strong>POST method</strong></td>
<td>Cette case est cochée par défaut, elle permet d'utiliser la méthode POST. Dans le tableau <strong>Parameters</strong>, saisissez le nom de la ou des variable(s) dans le champ <strong>Name</strong> et leur valeur correspondante dans le champ <strong>Value</strong>.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Décochez cette case pour passer les lignes en erreur et terminer le traitement des lignes sans erreur.</td>
</tr>
<tr>
<td><strong>Read Cookie</strong></td>
<td>Cochez cette case pour que le <strong>tFileFetch</strong> charge un cookie d'authentification d'une page Web.</td>
</tr>
<tr>
<td><strong>Save Cookie</strong></td>
<td>Cochez cette case pour sauvegarder le cookie d'authentification Web. Cela signifie que vous n'aurez plus à vous identifier sur le même site Web.</td>
</tr>
<tr>
<td><strong>Cookie file</strong></td>
<td>Saisissez le chemin d'accès complet au fichier à utiliser pour sauvegarder les cookies. Cliquez sur le bouton [...] et parcourrez votre système jusqu'au fichier dans lequel sauvegarder votre cookie.</td>
</tr>
<tr>
<td><strong>Cookie policy</strong></td>
<td>Sélectionnez une politique relative aux cookies dans la liste déroulante. Quatre options sont disponibles : <strong>BROWSER_COMPATIBILITY</strong>, <strong>DEFAULT</strong>, <strong>NETSCAPE</strong> et <strong>RFC_2109</strong>.</td>
</tr>
<tr>
<td>**Single cookie</td>
<td>Cochez cette case afin de rassembler tous les cookies dans l'en-tête de la requête afin d'obtenir une compatibilité maximum entre les différents serveurs.</td>
</tr>
<tr>
<td>header**</td>
<td>Cochez cette case afin de rassembler tous les cookies dans l'en-tête de la requête afin d'obtenir une compatibilité maximum entre les différents serveurs.</td>
</tr>
<tr>
<td><strong>Advanced settings</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Advanced settings**
### Timeout

Nombre de millisecondes avant fermeture de la connexion au protocole.
Disponible pour les protocoles **http** et **https**.

### Print response to console

Cochez cette case pour imprimer la réponse du serveur dans la console.
Disponible pour les protocoles **http** et **https**.

### Upload file

Cochez cette case pour charger un ou plusieurs fichier(s) sur le serveur. Puis, dans la table **Files** qui s’affiche, cliquez sur le bouton [+] afin d’ajouter le (les) fichier(s) à charger et définissez les paramètres suivant pour chaque fichier :

- **Name** : le nouveau du fichier après chargement, entre guillemets doubles.
- **File** : le chemin complet du fichier à charger, par exemple "D:/filefetch.txt".
- **Content-Type** : le type de contenu du fichier à charger. La valeur par défaut est "application/octet-stream".
- **Charset** : l’ensemble de caractères du fichier à charger. La valeur par défaut est "ISO-8859-1".

Disponible pour les protocoles **http** et **https**.

### Enable proxy server

Cochez cette case si vous vous connectez derrière un proxy et renseignez les informations correspondantes.
Disponible pour les protocoles **http**, **https** et **ftp**.

### Enable NTLM Credentials

Cochez cette case si vous utiliser un protocole d’authentification NTLM.

- **Domain** : Nom de domaine du client.
- **Host** : Adresse IP du client.

Disponible pour les protocoles **http**, **https** et **ftp**.

### Need authentication

Cochez la case d’authentification et renseignez le nom de l’utilisateur dans le champs **Username** et son mot de passe dans le champs **Password**, si cela est nécessaire pour accéder au protocole.

Disponible pour les protocoles **http** et **https**.

### Support redirection

Cochez cette case pour répéter la requête de redirection jusqu’à ce que la redirection soit effective et que le fichier puisse être récupéré.

Disponible pour les protocoles **http**, **https** et **ftp**.

### Global Variables

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères.
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**INPUT_STREAM** : le contenu du fichier récupéré. Cette variable est une variable **Flow** et retourne une classe InputStream.

Une variable **Flow** fonctionne durant l'exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en composant de départ pour alimenter le flux d’entrée d’un Job et est souvent utilisé avec un lien de type <strong>OnSubjobOk</strong> ou <strong>OnComponentOk</strong>, en fonction du contexte.</th>
</tr>
</thead>
</table>

| Limitation | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center ([https://help.talend.com](https://help.talend.com)). |

**Scénario 1 : Récupérer des données à partir d’un protocole HTTP**

Ce scénario décrit un Job comprenant trois composants, qui récupère un fichier d’un site internet HTTP, lit des données du fichier récupéré et affiche les données dans la console.
Déposer et relier les composants

Procédure
1. De la Palette, déposez les composants **tFileFetch**, **tFileInputDelimited** et **tLogRow** dans l’espace de modélisation graphique.
2. Reliez le **tFileFetch** au **tFileInputDelimited** à l’aide d’un lien **Trigger > OnSubjobOk** ou **OnComponentOk**.
3. Reliez le **tFileInputDelimited** au **tLogRow** à l’aide d’un lien **Row > Main**.

Configurer les composants

Procédure
1. Double-cliquez sur le **tFileFetch** pour ouvrir sa vue **Basic settings**.

   ![Basic settings](image)

2. Sélectionnez le protocole que vous souhaitez utiliser. Ici, utilisez le protocole **http**.
4. Dans le champ **Destination directory**, sélectionnez le répertoire dans lequel le fichier récupéré doit être stocké. Dans cet exemple, saisissez **D:/Output**.
5. Dans le champ **Destination filename**, saisissez le nouveau nom du fichier si vous souhaitez le renommer. Dans cet exemple, **new.txt**.
6. Si nécessaire, cochez la case **Add header** et définissez une ou plusieurs requêtes HTTP comme conditions de récupération. Par exemple, vous pouvez récupérer le fichier s’il a été modifié depuis une date précise.


7. Double-cliquez sur le **tFileInputDelimited** pour ouvrir sa vue **Basic settings**.

8. Dans le champ **File name**, saisissez le chemin d’accès complet au fichier récupéré stocké localement.

9. Cliquez sur le bouton […] à côté du champ **Edit schema** pour ouvrir la boîte de dialogue [Schema].

Dans cet exemple, ajoutez une colonne **output** pour stocker les données du fichier récupéré.

10. Laissez les autres paramètres tels qu’ils sont.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur F6 ou cliquez sur le bouton Run de la vue Run pour l’exécuter.

Starting job test at 17:14 10/01/2014.

[statistics] connecting to socket on port 3656
[statistics] connected
Well done
[statistics] disconnected
Job test ended at 17:14 10/01/2014. [exit code=0]

Les données du fichier récupéré s’affichent dans la console.

Scénario 2 : Réutiliser un cookie stocké pour récupérer des fichiers via un protocole HTTP

Ce scénario décrit un Job comprenant deux composants, qui se connecte à un site Web HTTP donné et, à l’aide d’un cookie stocké dans un répertoire local spécifié, récupère les données de ce site.

Déposer et relier les composants

Procédure
1. Déposez deux composants tFileFetch de la Palette dans l’espace de modélisation graphique.
2. Reliez ces deux composants (en tant que sous-jobs) à l’aide d’un lien Trigger > OnSubjobOk.

Configurer les composants

Configurer le premier sous-job

Procédure
1. Double-cliquez sur le tFileFetch_1 afin d’ouvrir sa vue Component et définir ses propriétés.
2. Dans la liste **Protocol**, sélectionnez le protocole que vous souhaitez utiliser. Ici, utilisez le protocole **https**.


4. Dans le champ **Destination directory**, parcourez votre répertoire jusqu'au dossier dans lequel stocker la page Web récupérée. Ce dossier sera créé à la volée s'il n'existe pas. Dans cet exemple, saisissez **D:/download**.

5. Dans le champ **Destination filename**, saisissez un nouveau nom pour la page Web si vous souhaitez le modifier. Dans cet exemple, saisissez **codeproject.html**.

6. Dans le tableau **Parameters**, cliquez deux fois sur le bouton **[+]** pour ajouter deux lignes et renseignez les informations d'authentification pour accéder au site Web souhaité.

   En première colonne **Name**, saisissez respectivement les noms cyan **Email** et **Password**, requises par le site Web auquel vous vous connectez.

   En deuxième colonne **Value**, saisissez respectivement votre adresse e-mail et votre mot de passe.

7. Cochez la case **Save cookie**.

8. Dans le champ **Cookie file**, saisissez le chemin d'accès complet au fichier dans lequel vous souhaitez stocker le cookie. Dans cet exemple, le chemin est **D:/download/cookie**.

9. Cliquez sur l'onglet **Advanced settings** pour ouvrir cette vue.

10. Cochez la case **Support redirection** afin que la requête de redirection soit répétée jusqu'à ce que la redirection réussisse.
Configurer le second sous-job

Procédure
1. Double-cliquez sur le tFileFetch_2 afin d'ouvrir sa vue Component et définir ses propriétés.

2. Dans la liste Protocol, sélectionnez http.
4. Dans le champ Destination directory, parcourez votre répertoire jusqu'au dossier dans lequel vous souhaitez stocker les fichiers récupérés. Ce dossier peut être automatiquement créé s'il n'existe pas durant le processus d'exécution. Dans cet exemple, saisissez D:/download.
5. Dans le champ Destination filename, saisissez un nouveau nom pour le fichier si vous souhaitez le modifier. Dans cet exemple, saisissez source.zip.
6. Décochez la case POST method pour désactiver le tableau Parameters.
7. Cochez la case Read cookie.
8. Dans le champ Cookie file, parcourez votre système jusqu'au fichier utilisé pour sauvegarder le cookie. Dans cet exemple, le fichier est D:/download/cookie.

Sauvegarder et exécuter le Job

Procédure
1. Sauvegardez votre Job en appuyant sur les touches Ctrl+S.
2. Appuyez sur F6 pour l'exécuter ou cliquez sur le bouton Run dans l'onglet Run.
   Allez dans le dossier local D:/download pour vérifier le fichier téléchargé.
Scénario associé

Pour un exemple de transfert de données en mode stream, consultez Scénario 2 : Lire les données d'un fichier distant en mode stream à la page 1072.
tFileInputARFF

Ce composant lit un fichier ARFF ligne par ligne afin de le diviser en champs et d’envoyer ces champs au composant suivant, comme défini par le schéma.

Propriétés du tFileInputARFF Standard

Ces propriétés sont utilisées pour configurer le tFileInputARFF s’exécutant dans le framework de Jobs Standard.
Le composant tFileInputARFF Standard appartient à la famille File.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Built-in : Propriétés utilisées ponctuellement.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les</td>
</tr>
</tbody>
</table>
modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].


### Advanced settings

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les informations de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Utilisez ce composant pour lire un fichier et séparer les champs à l’aide du séparateur spécifié.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton Install dans</td>
</tr>
</tbody>
</table>
Scénario : Afficher le contenu d’un fichier ARFF

Le scénario suivant décrit un Job formé de deux composants ayant pour objectif de lire ligne à ligne un fichier ARFF, de sélectionner des données délimitées et d’afficher la sortie dans la console de la vue Run.

Un fichier ARFF se présente comme suit :

```plaintext
@relation vote
@attribute 'handicapped-infants' {'n', 'y'}
@attribute 'water-project-cost-sharing' {'n', 'y'}
@attribute 'adoption-of-the-budget-resolution' {'n', 'y'}
@attribute 'physician-fee-freeze' {'n', 'y'}
@attribute 'el-salvador-aid' {'n', 'y'}
@attribute 'religious-groups-in-schools' {'n', 'y'}
@attribute 'anti-satellite-test-ban' {'n', 'y'}
@attribute 'aid-to-nicaraguan-contras' {'n', 'y'}
@attribute 'mx-missile' {'n', 'y'}
@attribute 'immigration' {'n', 'y'}
@attribute 'synfuels-corporation-cutback' {'n', 'y'}
@attribute 'education-spending' {'n', 'y'}
@attribute 'superfund-right-to-sue' {'n', 'y'}
@attribute 'crime' {'n', 'y'}
@attribute 'duty-free-exports' {'n', 'y'}
@attribute 'export-administration-act-south-africa' {'n', 'y'}
@attribute 'class' {'democrat', 'republican'}
@data
'n','y','n','y','y','y','n','n','n','y','?','y','y','y','n','n','y','republican'
'n','y','n','y','y','y','n','n','n','n','n','?','y','y','y','n','n','?','republican'
'y','y','?','y','y','y','n','n','n','n','y','y','y','n','n','democrat'
```

Un fichier ARFF est généralement composé de deux parties : la description de la structure des données, correspondant aux lignes qui débutent par `@attribute`, et les données brutes, qui suivent l’expression `@data`.

Déposer et relier les composants

**Procédure**

1. Glissez et déposez le composant `tFileInputARFF` de la Palette dans l’espace de modélisation graphique.
2. Glissez et déposez de la même manière le composant `tLogRow`
3. Cliquez-droit sur le composant **tFileInputARFF** et sélectionnez **Row > Main** dans le menu contextuel. Glissez ce lien vers le **tLogRow** et relâchez lorsque le symbole de prise de courant apparaît. La transmission du flux de donnée entre les deux composants est ainsi établie.

###Configurer les composants

####Procédure

1. Double-cliquez sur le **tFileInputARFF** pour paramétrer ce composant.
2. Dans la vue **Component**, renseignez le champ **File Name**. Parcourez votre répertoire de fichiers, afin de sélectionner le fichier .arff.
3. Dans le champ **Schema**, sélectionnez **Built-In**.
5. Cliquez autant de fois sur le bouton que nécessaire pour décrire chacune des colonnes du fichier lu. Dans cet exemple, 17 colonnes sont ajoutées, nommez-les comme suit :

![Schema of tFileInputARFF_1](image)

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Type</th>
<th>N..</th>
<th>Dat..</th>
<th>L..</th>
<th>Pre..</th>
<th>D..</th>
</tr>
</thead>
<tbody>
<tr>
<td>HandicappedInfants</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WaterProjectCostSharing</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AdoptionOfTheBudgetResolution</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PhysicianFeeFreeze</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESalvadorAid</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReligiousGroupsInSchools</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AntiBasiliteTestBan</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AdToNicaraguanContras</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MxMissile</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Immigration</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SynfuelCorporationCutback</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EducationSpending</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SupercostRightToSue</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crime</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DutyFreeExports</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExportAdministrationActSouthAfrica</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClassName</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. Pour chacune des colonnes sélectionnez le **Type** : chaîne de caractères (**String**).
7. Cliquez sur **OK**.
8. Dans l’espace de modélisation, double-cliquez sur le composant **tLogRow** pour afficher sa vue **Component**.
9. Cliquez sur le bouton [...] du champ **Edit schema** pour vérifier que le schéma a bien été propagé. Si ce n’est pas le cas, cliquez sur le bouton **Sync columns**.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** pour l’exécuter.

**Résultats**

La console affiche les données contenues dans le fichier ARFF délimitées par une barre verticale (separateur par défaut).
tFileInputDelimited

Ce composant lit un fichier délimité ligne par ligne, afin de le diviser en champs et d’envoyer ces champs au composant suivant, comme défini par le schéma.

Propriétés du tFileInputDelimited Standard

Ces propriétés sont utilisées pour configurer le tFileInputDelimited s’exécutant dans le framework de Jobs Standard.

Le composant tFileInputDelimited Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-In : propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
</tbody>
</table>

**File Name/Stream**

File name : Chemin d’accès et nom du fichier à traiter.

Stream : Flux de données à traiter. Les données doivent préalablement être mises en flux afin d’être récupérées par le tFileInputDelimited via la variable représentative correspondante.

Cette variable peut être prédéfinie dans votre Studio ou fournie par le contexte ou les composants utilisé(s) avec ce composant. Sinon, vous pouvez la définir manuellement et l’utiliser selon votre Job, par exemple à l’aide d’un tJava ou d’un tJavaFlex.

Afin d’éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d’autocomplétion (Ctrl+Espace) afin de remplir le champ, si cette variable a été correctement définie.

Pour plus d’informations concernant les variables disponibles, consultez le Guide utilisateur du Studio Talend.

**Row separator**

Saisissez le séparateur utilisé pour identifier la fin d’une ligne.

**Field separator**

Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.

**CSV options**

Cochez cette case afin de spécifier les paramètres CSV suivants :

- **Escape char** : saisissez le caractère d’échappement entre guillemets doubles. Par exemple, vous devez
saisir "\" si le caractère barre oblique inversée (\) est utilisé comme caractère d'échappement.

- **Text enclosure** : saisissez le caractère avec lequel entourer le texte (un seul caractère) entre guillemets doubles. Par exemple, vous devez saisir "***" lorsque le guillemet double (") est utilisé comme caractère pour entourer le texte.

<table>
<thead>
<tr>
<th><strong>Header</strong></th>
<th>Saisissez le nombre de lignes à ignorer au début du fichier.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Footer</strong></td>
<td>Nombre de lignes à ignorer à la fin d'un fichier.</td>
</tr>
<tr>
<td><strong>Limit</strong></td>
<td>Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n'est lue ni traitée.</td>
</tr>
</tbody>
</table>

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Notez que, si la valeur d'entrée d'un champ primitif non nullable est nulle, la ligne de données comprenant ce champ sera rejetée.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


**Skip empty rows** Cochez cette case pour ignorer les lignes vides.

**Uncompress as zip file** Cochez cette case pour décompresser le fichier d'entrée.

**Die on error** Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient.
<table>
<thead>
<tr>
<th><strong>Advanced settings</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advanced separator (for numbers)</strong></td>
<td>Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers est une virgule (,) et le séparateur décimal est un point (.)</td>
</tr>
<tr>
<td><strong>Extract lines at random</strong></td>
<td>Cochez cette case pour définir le nombre de lignes à extraire de façon aléatoire</td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.</td>
</tr>
<tr>
<td><strong>Trim all column</strong></td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées. Lorsque cette case est cochée, la table <strong>Check column to trim</strong> est affichée, ce qui vous permet de sélectionner les colonnes desquelles vous souhaitez enlever les espaces en début et en fin de champ.</td>
</tr>
<tr>
<td><strong>Check each row structure against schema</strong></td>
<td>Cochez cette case afin de vérifier que le nombre total de colonnes est cohérent par rapport au schéma. Si le nombre n’est pas cohérent, un message d’erreur s’affiche dans la console.</td>
</tr>
<tr>
<td><strong>Check date</strong></td>
<td>Cochez cette case pour vérifier strictement le format de date par rapport au schéma d’entrée.</td>
</tr>
<tr>
<td><strong>Check columns to trim</strong></td>
<td>Cette table est automatiquement renseignée avec le schéma utilisé. Cochez la (les) case(s) correspondant à la (aux) colonne(s) dont vous souhaitez supprimer les espaces et début et en fin de champ.</td>
</tr>
<tr>
<td><strong>Split row before field</strong></td>
<td>Cochez cette case pour séparer les lignes avant de séparer les champs.</td>
</tr>
<tr>
<td><strong>Permit hexadecimal (0xNNN) or octal (0NNNN) for numeric types</strong></td>
<td>Cochez cette case si l’un de vos types numériques (long, integer, short, ou byte), doit être parsé depuis une chaîne de caractères hexadécimale ou octale. Cochez cette case à côté de la ou des colonne(s) d’intérêt pour transformer la chaîne de caractères d’entrée de chaque colonne sélectionnée en un type de données qui est celui défini dans le schéma. Cochez la case <strong>Permit hexadecimal or octal</strong> pour sélectionner toutes les colonnes.</td>
</tr>
</tbody>
</table>
Cette table apparaît uniquement lorsque la case Permit hexadecimal (0xNNN) or octal (0NNNN) for numeric types est cochée.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

**Règle d’utilisation**

Utilisez ce composant pour lire un fichier et séparer les champs à l’aide du séparateur spécifié. Il permet de créer un flux de données à l’aide d’un lien Row > Main, ainsi que de créer un flux de rejet avec un lien Row > Reject filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d’utilisation de ces deux liens, consultez Procédure à la page 1152 du composant tFileInputXML.

**Limitation**


### Scénario 1 : Afficher le contenu d'un fichier délimité

Le scénario suivant est un Job de deux composants ayant pour objectif de lire les lignes d’un fichier, de sélectionner des données délimitées et d’afficher la sortie dans la console de la vue Run.
Déposer et relier les composants

Procédure
1. Cliquez et déposez un composant **tFileInputDelimited** et un composant **tLogRow** de la Palette dans l’espace de modélisation.
2. Cliquez-droit sur le composant **tFileInputDelimited** et sélectionnez Row > Main dans le menu contextuel. Puis glissez ce lien vers le **tLogRow** et relâchez lorsque le symbole de prise de courant apparaît.

Configurer les composants

Procédure
1. Sélectionnez le composant **tFileInputDelimited** à nouveau et définissez ses propriétés dans l’onglet Basic settings :

![Configuration de tFileInputDelimited](image)

2. Renseignez le chemin d’accès au fichier dans le champ **File Name**. Ce champ est obligatoire.

   **Avertissement :**

   Si le chemin d’accès du fichier contient des caractères accentués, vous obtiendrez un message d’erreur lors de l’exécution du Job.

3. Définissez le séparateur de lignes dans le champ **Row separator** afin d’identifier la fin d’une ligne. Puis définissez le séparateur de champs dans **Field separator** pour délimiter les champs composant une ligne.

4. Dans ce scénario, l’en-tête (**Header**) et le pied de page (**Footer**) n’ont pas besoin d’être définis. Et la limite de lignes lues (**Limit**) est de 50 pour cet exemple.

5. Sélectionnez soit local (**Built-in**) soit distant (**Repository**) comme **Schema** pour définir les données qui passent par le composant **tLogRow**.


8. Sélectionnez le composant tLogRow et définissez le séparateur de champs de la sortie affichée. Voir également Propriétés du tLogRow Standard à la page 2105.

9. Cochez la case Print schema column name in front of each value pour récupérer le libellé des colonnes dans la sortie affichée.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur Ctrl+S pour sauvegarder votre Job.

2. Cliquez sur la vue Run, puis cliquez sur Run pour exécuter le Job.

Le fichier est lu ligne par ligne et les champs extraits sont affichés dans la console, tel que défini dans les propriétés du composant (dans l’onglet Basic settings).

<table>
<thead>
<tr>
<th>ID_Owner</th>
<th>Name</th>
<th>ID_Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>lebouch</td>
<td>PKI2906</td>
</tr>
<tr>
<td>3</td>
<td>hookman</td>
<td>ENU9147</td>
</tr>
<tr>
<td>4</td>
<td>kirtle</td>
<td>TEV8360</td>
</tr>
<tr>
<td>5</td>
<td>bobouh</td>
<td>VFM3151</td>
</tr>
<tr>
<td>6</td>
<td>carbone</td>
<td>IFY9035</td>
</tr>
</tbody>
</table>

**Résultats**

La console affiche l’en-tête suivi des données lues par le Job.

**Scénario 2 : Lire les données d’un fichier distant en mode stream**

Le scénario suivant illustre un Job à quatre composants ayant pour objectif de récupérer les données d’un fichier volumineux quasiment aussitôt qu’elles ne sont lues afin d’éviter l’attente du téléchargement de l’ensemble du fichier de données, et les affichent dans la console de la vue Run.
Déposer et relier les composants

Procédure
1. Déposez les composants suivants dans l’espace de modélisation : tFileFetch, tSleep, tFileInputDelimited, et tLogRow.
2. Reliez les composants tSleep et tFileInputDelimited via un lien de type Trigger > OnComponentOk et reliez les composants tFileInputDelimited et tLogRow via un lien de type Row > Main.

Configurer les composants

Procédure
1. Double-cliquez sur le composant tFileFetch pour afficher l’onglet Basic settings de la vue Component et paramétrer ses propriétés.

   ![tFileFetch](Image)
   - **Basic settings**
     - **Protocol** : http
     - **URI** : "http://www.talend.com/US_Employees.csv"
     - **Use cache to save the resource**
     - **POST method**
     - **Die on error**
     - **Parameters**

   - **View**
   - **Documentation**

2. Dans la liste Protocol, sélectionnez le type de protocole vous permettant d’accéder au serveur sur lequel est stocké votre fichier de données.
3. Dans le champ URI, saisissez l’URI d’accès au serveur sur lequel est stocké votre fichier.
4. Cochez la case **Use cache to save the resource** pour mettre les données de votre fichier en mémoire cache. Cette option permet d’utiliser le mode stream pour transférer les données du fichier.
5. A partir de l’espace de modélisation, double-cliquez sur le composant tSleep pour afficher l’onglet Basic settings de la vue Component et paramétrer ses propriétés.
   Par défaut, le champ **Pause** du composant tSleep correspond à 1 seconde. Laissez ce paramètre par défaut. Il met le deuxième Job en pause afin de laisser le temps au premier Job contenant le tFileFetch de lire les données du fichier.
6. A partir de l’espace de modélisation, double-cliquez sur le composant tFileInputDelimited pour afficher l’onglet Basic settings de la vue Component et paramétrer ses propriétés.
7. Dans le champ **File name/Stream** :
   - Effacez le contenu par défaut.
   - Faites un **Ctrl+Espace** pour faire apparaître la liste des variables disponibles à partir de ce composant.
   - Sélectionnez la variable **tFileFetch_1_INPUT_STREAM** dans la liste d'auto-complétion, la variable suivante apparaît dans le champ **Filename** : `((java.io.InputStream)globalMap.get("tFileFetch_1_INPUT_STREAM"))`.

8. Dans la liste **Schema**, sélectionnez **Built-in** et cliquez sur le bouton `[...]` à côté du champ **Edit schema** pour décrire la structure des données du fichier que vous souhaitez récupérer. Le fichier **US_Employees** est composé de six colonnes : **ID, Employee, Age, Address, State, EntryDate**.
   Cliquez sur le bouton `[+]` pour ajouter les six colonnes et paramétrez-les comme indiqué sur la capture d'écran ci-dessus. Cliquez sur **OK**.

9. A partir de l'espace de modélisation, double-cliquez sur le composant **tLogRow** pour afficher l'onglet **Basic settings** de la vue **Component** et cliquez sur **Sync Columns** pour vous assurer que le schéma est récupéré du composant précédent.
Configurer l’exécution du Job et exécuter le Job

Procédure


2. Cochez la case Multi thread execution afin d’exécuter les deux Jobs en parallèle, sachant que le deuxième Job a un retard d’une seconde par rapport au premier comme défini dans le tSleep. Cette option permet donc de récupérer les données grâce au tFileDelimited quasiment dès leur lecture par le tFileFetch.


Les données sont affichées dans la console dès leur lecture.
tFileInputExcel

Ce composant lit un fichier Excel ligne par ligne pour le scinder en champs et envoie les champs comme défini dans le schéma au composant suivant.

Propriétés du tFileInputExcel Standard

Ces propriétés sont utilisées pour configurer le tFileInputExcel s’exécutant dans le framework de Jobs Standard.
Le composant tFileInputExcel Standard appartient à la famille File.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Cliquez sur cette icône pour ouvrir l’assistant de connexion et enregistrer les paramètres de connexion au fichier Excel que vous avez définis dans la vue Basic settings du composant.

Pour plus d’informations sur comment définir et stocker des paramètres de connexion de fichier, consultez le Guide utilisateur du Studio Talend.

Read excel2007 file format (xlsx)


File name/Stream

File name : Chemin d’accès et nom du fichier, et/ou variable à traiter.
Stream : Flux de données à traiter. Les données doivent préalablement être mises en flux afin d’être récupérées par le tFileInputExcel via la variable INPUT_STREAM disponible dans la liste d’auto-complétion (Ctrl+Espace).

Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend.

All sheets

Cochez cette case pour traiter toutes les feuilles du fichier Excel.

Sheet list

Cliquez sur le bouton [+] pour ajouter de nouvelles lignes à la liste de feuilles à traiter :
Sheet (name or position) : renseignez le nom ou la position de la feuille à traiter.
<table>
<thead>
<tr>
<th><strong>Use Regex</strong></th>
<th>Cochez cette case si vous utilisez une expression régulière pour filtrer les feuilles selon un modèle.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Header</strong></td>
<td>Saisissez le nombre de lignes à ignorer au début du fichier.</td>
</tr>
<tr>
<td><strong>Footer</strong></td>
<td>Nombre de lignes à ignorer à la fin d’un fichier.</td>
</tr>
<tr>
<td><strong>Limit</strong></td>
<td>Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n’est lue ni traitée.</td>
</tr>
<tr>
<td><strong>Affect each sheet(header&amp;footer)</strong></td>
<td>Cochez cette case pour appliquer les paramètres définis dans les champs Header et Footer à toutes les feuilles traitées.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Reject.</td>
</tr>
<tr>
<td><strong>First column et Last column</strong></td>
<td>Renseignez l’intervalle de colonnes que vous voulez traiter : la position de la première colonne dans le champ First column et celle de la dernière colonne à traiter dans le champ Last column.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
## Advanced settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Advanced separator (for numbers)**        | Cochez cette option pour modifier les séparateurs utilisés pour les nombres :  
- **Thousands separator** : définissez le séparateur utilisé pour les milliers.  
- **Decimal separator** : définissez le séparateur utilisé pour les décimaux. |
| **Trim all columns**                         | Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.  
Lorsque cette case est cochée, la table **Check column to trim** est affichée, ce qui vous permet de sélectionner les colonnes desquelles vous souhaitez enlever les espaces en début et en fin de champ. |
| **Check column to trim**                     | Cette table est automatiquement renseignée avec le schéma utilisé. Cochez la (les) case(s) correspondant à la (aux) colonne(s) dont vous souhaitez supprimer les espaces et début et en fin de champ. |
| **Convert date column to string**            | Disponible lorsque la case **Read excel2007 file format (xlsx)** est cochée, dans la vue **Basic settings**.  
Cochez cette case pour afficher la table **Check need convert date column**. Vous pouvez convertir les colonnes Excel contenant des dates en colonnes de type String dans le fichier de sortie, avec le modèle défini.  
- **Column** : toutes les colonnes disponibles dans le schéma du fichier source .xlsx.  
- **Convert** : cochez cette case pour choisir toutes les colonnes à convertir (si elles sont de type String). Vous pouvez également cocher la case de chaque colonne à convertir.  
- **Date pattern** : définissez le format de date. |
| **Encoding**                                 | Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données. |
| **Read real values for numbers**             | Cochez cette case pour lire les nombres en valeur réelle. Cette case est indisponible lorsque vous cochez la case **Read excel2007 file format (xlsx)** dans la vue **Basic settings**. |
| **Stop reading on encountering empty rows**  | Cochez cette case pour ignorer les lignes vides, et, s’il y en a, les lignes qui suivent. Cette case est indisponible lorsque vous cochez la case **Read excel2007 file format (xlsx)** dans la vue **Basic settings**. |
| **Generation mode**                          | Disponible lorsque la case **Read excel2007 file format (xlsx)** est cochée dans la vue **Basic settings**.  
Sélectionnez le mode de lecture du fichier Excel 2007.  
- **Less memory consumed for large excel(Event mode)** : utilisé pour un fichier volumineux. Ce mode
économise de la mémoire car il permet de lire le fichier Excel 2007 comme un flux. Lorsque ce mode est sélectionné, les données seront extraites avec le symbole du format, par exemple le symbole de pourcentage % et le symbole de devise $.

- **Memory-consuming (User mode)** : utilisé pour un petit fichier. Nécessite beaucoup de mémoire. Lorsque ce mode est sélectionné, les données pures sans le symbole de format seront extraites.

<table>
<thead>
<tr>
<th>Don't validate the cells</th>
<th>Cochez cette case pour ne pas effectuer de validation des données. Cette case est indisponible lorsque vous cochez la case <em>Read excel2007 file format (xlsx)</em> dans la vue <em>Basic settings</em>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore the warning</td>
<td>Cochez cette case pour ignorer les avertissements générés par des erreurs sur le fichier Excel. Cette case est indisponible lorsque vous cochez la case <em>Read excel2007 file format (xlsx)</em> dans la vue <em>Basic settings</em>.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>NB_LINE</strong> : nombre de lignes traitées. Cette variable est une variable <em>After</em> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>CURRENT_SHEET</strong> : nom de la feuille traitée. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable *Flow* fonctionne durant l'exécution d'un composant. Une variable *After* fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Utilisez ce composant pour lire un fichier Excel (.xls ou .xlsx) et séparer les champs en fonction d'un schéma défini. Ce composant permet de créer un flux de données à l'aide d'un lien <em>Row &gt; Main</em>, ainsi que de créer un flux de rejet avec un lien <em>Row &gt; Reject</em> filtrant les données dont le type ne correspond pas au type</th>
</tr>
</thead>
</table>
défini. Pour un exemple d’utilisation de ces deux liens, consultez Procédure à la page 1152 du composant tFileInputXML.

<table>
<thead>
<tr>
<th>Limitation</th>
</tr>
</thead>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tFileInputFullRow

Ce composant lit un fichier ligne par ligne et envoie les lignes de données complètes au composant suivant comme défini dans le schéma, via une connexion de type Row.

Propriétés du tFileInputFullRow Standard

Ces propriétés sont utilisées pour configurer le tFileInputFullRow s'exécutant dans le framework de Jobs Standard.

Le composant tFileInputFullRow Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

| **File Name** | Spécifiez le chemin d’accès du fichier à traiter. |
| **Row separator** | Saisissez le séparateur utilisé pour identifier la fin d’une ligne. |
| **Header** | Saisissez le nombre de lignes à ignorer au début du fichier. |

Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Footer
Saisissez le nombre de lignes à ignorer à la fin du fichier.

Limit
Saisissez le nombre maximum de lignes à traiter. Si la valeur est fixée à 0, aucune ligne n’est lue ou traitée.

Skip empty rows
Cochez cette case pour ignorer les lignes vides.

Advanced settings

Encoding
Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.

Extract lines at random
Cochez cette case pour définir le nombre de lignes à extraire de façon aléatoire.

tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant.

Global Variables

Global Variables

NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation
Utilisez ce composant pour lire des lignes complètes dans un fichier délimité pouvant être très gros.

Scénario : Lire des lignes complètes dans un fichier délimité

Le scénario suivant crée un Job à deux composants permettant de lire des lignes complètes à partir du fichier délimité states.csv et d’afficher les données dans la console.
Le contenu du fichier states.csv qui contient dix lignes de données se présente comme suit :

StateID;StateName
1;Alabama
2;Alaska
3;Arizona
4;Arkansas
5;California
6;Colorado
7;Connecticut
8;Delaware
9;Florida
10;Georgia

**Lire des lignes complètes dans un fichier délimité**

**Procédure**

1. Créer un nouveau Job et ajouter un tFileInputFullRow et un tLogRow en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.
2. Relier le tFileInputFullRow au tLogRow à l’aide d’un lien Row > Main.


4. Cliquez sur le bouton [...] du champ Edit schema pour voir les données à transférer au composant tLogRow. Notez que le schéma est en lecture seule et ne comporte qu’une seule colonne, line.
5. Dans le champ **File Name**, parcourez votre système jusqu’au fichier à traiter ou saisissez son chemin d'accès. Dans ce scénario, le fichier est *E:/states.csv*.

6. Dans le champ **Row separator**, renseignez le séparateur permettant d'identifier les fins de ligne. Dans cet exemple, la valeur par défaut est 

7. Dans le champ **Header**, indiquez la valeur 1 pour ignorer les lignes d'en-tête au début du fichier.

8. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings** dans l'onglet **Component**.

Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour afficher un résultat plus lisible.

9. Appuyez sur **Ctrl+S** pour sauvegarder votre Job puis sur **F6** pour l’exécuter.

```plaintext
[statistics] connecting to socket on port 3617
[statistics] connected
| tLogRow_1
|-----------------
| line
| ===========
| 1: Alabama
| 2: Alaska
| 3: Arizona
| 4: Arkansas
| 5: California
| 6: Colorado
| 7: Connecticut
| 8: Delaware
| 9: Florida
| 10: Georgia
|-----------------
[statistics] disconnected
```

Comme affiché ci-dessus, les dix lignes de données du fichier *states.csv* sont lues une par une, en ignorant les séparateurs de champs, et les lignes de données complètes sont affichées dans la console.

Pour extraire les champs des lignes, vous devez utiliser un **tExtractDelimitedFields**, un **tExtractPositionalFields** ou un **tExtractRegexFields**. Pour plus d'informations, consultez **tExtractDelimitedFields** à la page 993, **tExtractPositionalFields** à la page 1014 et **tExtractRegexFields** à la page 1017.
tFileInputJSON

Ce composant extrait des données JSON d’un fichier et transfère les données à un fichier, une table de base de données, etc.

Propriétés du tFileInputJSON Standard

Ces propriétés sont utilisées pour configurer le tFileInputJSON s’exécutant dans le framework de Jobs Standard.
Le composant tFileInputJSON Standard appartient aux familles Internet et File.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-In : propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
</tbody>
</table>

Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

- Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur Edit schema pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.

- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
<table>
<thead>
<tr>
<th><strong>Read By</strong></th>
<th>Sélectionnez un moyen d'extraire les données JSON du fichier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>JsonPath</strong> : extrait les données JSON en se basant sur la requête JSONPath. Lorsque cette option est sélectionnée, vous devez sélectionner une version de l'API JSONPath dans la liste <strong>API version</strong>. Il est recommandé de lire les données via JSONPath afin d'améliorer les performances.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Xpath</strong> : extrait les données JSON en se basant sur la requête XPath.</td>
<td></td>
</tr>
<tr>
<td>• <strong>JsonPath without loop</strong> : extrait les données JSON en se basant sur la requête JSONPath sans configurer de nœud de boucle.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Use Url</strong></th>
<th>Cochez cette case pour récupérer les données directement à partir d'internet.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>URL</strong></th>
<th>Saisissez le chemin de l'URL d'où vous souhaitez récupérer les données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce champ est disponible uniquement lorsque la case <strong>Use Url</strong> est cochée.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Filename</strong></th>
<th>Spécifiez le fichier duquel récupérer les données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce champ n'est pas disponible lorsque la case <strong>Use Url</strong> est cochée.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Loop Jsonpath query</strong></th>
<th>Spécifiez le nœud JSONPath sur lequel se base la boucle.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si vous avez sélectionné <strong>Xpath</strong> dans la liste déroulante <strong>Read by</strong>, le champ <strong>Loop Xpath query</strong> s'affiche.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Mapping</strong></th>
<th>Renseignez cette table pour mapper les colonnes définies dans le schéma aux nœuds JSON correspondants.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>Column</strong> : les cellules <strong>Column</strong> sont automatiquement renseignées avec le nom des colonnes définies dans le schéma.</td>
<td></td>
</tr>
<tr>
<td>Cette colonne est disponible uniquement lorsque l'option <strong>JsonPath</strong> est sélectionnée dans la liste <strong>Read By</strong>.</td>
<td></td>
</tr>
<tr>
<td>• <strong>XPath query</strong> : spécifie le nœud XPath contenant les données souhaitées.</td>
<td></td>
</tr>
<tr>
<td>Cette colonne est disponible uniquement lorsque l'option <strong>Xpath</strong> est sélectionnée dans la liste <strong>Read By</strong>.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Get Nodes</strong> : cochez cette case pour extraire les données JSON de tous les nœuds ou cochez la case à côté d'un nœud spécifique pour en extraire les données.</td>
<td></td>
</tr>
</tbody>
</table>
Cette colonne est disponible uniquement lorsque l’option **Xpath** est sélectionnée dans la liste **Read By**.

| **Die on error** | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**. |

**Advanced settings**

| **Advanced separator (for numbers)** | Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers est une virgule (,) et le séparateur décimal est un point (.).

**Thousands separator** : définissez le séparateur utilisé pour les milliers.

**Decimal separator** : définissez le séparateur utilisé pour les décimaux. |

| **Validate date** | Cochez cette case pour vérifier strictement le format de date par rapport au schéma d’entrée.
Cette case n’est disponible que si la case **Read By XPath** est cochée. |

| **Encoding** | Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données. |

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant. |

**Global Variables**

| **Global Variables** | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.  

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espacement** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**. |

1087
Scénario 1 : Extraire des données JSON d'un fichier en utilisant le JSONPath sans configurer de nœud de boucle

Ce scénario décrit un Job à deux composants qui extrait des données du fichier JSON Store.json en spécifiant le chemin JSON complet pour chaque nœud présentant un intérêt et affiche les données plates extraites dans la console.

Le fichier JSON contient des informations concernant un magasin et son contenu se présente comme suit :

```json
{"store": {
    "name": "Sunshine Department Store",
    "address": "Wangfujing Street",
    "goods": {
      "book": [
        {
          "category": "Reference",
          "title": "Sayings of the Century",
          "author": "Nigel Rees",
          "price": 8.88
        },
        {
          "category": "Fiction",
          "title": "Sword of Honour",
          "author": "Evelyn Waugh",
          "price": 12.66
        }
      ],
      "bicycle": {
        "type": "GIANT OCR2600",
        "color": "White",
        "price": 276
      }
    }
}}
```

Dans l'exemple suivant, vous allez extraire de ce fichier le nom du magasin, son adresse et les informations sur les vélos.
Ajouter et relier les composants

Procédure

1. Créez un nouveau Job et ajoutez un composant tFileInputJSON et un composant tLogRow en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le composant tFileInputJSON au tLogRow à l'aide d'un lien Row > Main.
3. Reliez les composants à l'aide d'un lien Row > Main.

Configurer les composants

Procédure

1. Double-cliquez sur le composant tFileInputJSON pour ouvrir sa vue Basic settings.

2. Sélectionnez JsonPath without loop dans la liste déroulante Read By. Pour cette option, vous devez spécifier le chemin JSON complet pour chaque nœud présentant un intérêt dans les champs JSONPath query de la table Mapping.

4. Cliquez sur le bouton [+] pour ajouter cinq colonnes, store_name, store_address, bicycle_type et bicycle_color de type String et bicycle_price de type Double.
   Cliquez sur OK pour fermer l'éditeur du schéma. Dans la fenêtre qui s'affiche, cliquez sur Yes pour propager le schéma au composant suivant.

5. Dans la boîte de dialogue [Propagate], cliquez sur Yes pour propager le schéma au composant suivant.
   Dans le champ Filename, spécifiez le chemin d'accès au fichier JSON contenant les données à extraire. Dans cet exemple, “E:/Store.json”.

6. Dans la table Mapping, les champs Column sont automatiquement renseignés par les colonnes du schéma que vous avez définies.
   Dans les champs JSONPath query, saisissez les requêtes JSONPath entre guillemets doubles pour spécifier les nœuds contenant les données souhaitées.
   - Pour les colonnes store_name et store_address, saisissez les expressions de requêtes JSONPath “$.store.name” et “$.store.address” relatives aux nœuds name et address, respectivement.
   - Pour les colonnes bicycle_type, bicycle_color et bicycle_price, saisissez les expressions de requêtes JSONPath “$.store.goods.bicycle.type”, “$.store.goods.bicycle.color” et “$.store.goods.bicycle.price” relatives aux nœuds enfants, respectivement type, color et price du nœud bicycle.

7. Double-cliquez sur le composant tLogRow pour afficher sa vue Basic settings.

8. Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour afficher un résultat plus lisible.
Exécuter le Job

Procédure
1. Appuyez sur les touches Ctrl+S pour sauvegarder le Job.
2. Appuyez sur F6 pour exécuter le Job.

Comme affiché ci-dessus, le nom du magasin, son adresse et les informations sur les vélos sont extraites des données source JSON et affichées sous forme de table plate dans la console.

Scénario 2 : Extraire des données JSON d’un fichier en utilisant JSONPath

Basé sur Scénario 1 : Extraire des données JSON d’un fichier en utilisant le JSONPath sans configurer de nœud de boucle à la page 1088, ce scénario montre comment extraire les données du tableau book du fichier JSON Store.json en spécifiant un nœud de boucle et le chemin JSON relatif pour chacun des nœuds recherchés, puis comment afficher les données extraites dans la console.

Procédure

Procédure
1. Dans le Studio, ouvrez le Job utilisé dans Scénario 1 : Extraire des données JSON d’un fichier en utilisant le JSONPath sans configurer de nœud de boucle à la page 1088 pour l’afficher dans l’espace de modélisation graphique.
2. Double-cliquez sur le composant tFileInputJSON pour ouvrir sa vue Basic settings.
3. Sélectionnez **JsonPath** dans la liste déroulante **Read By**.


5. Cliquez sur le bouton […] à côté de **Edit schema** pour ouvrir l’éditeur du schéma.

Sélectionnez les cinq colonnes ajoutées précédemment et cliquez sur le bouton [x] pour les supprimer toutes.

Cliquez sur le bouton [+] pour ajouter quatre colonnes, **book_title**, **book_category** et **book_author** de type **String** et **book_price** de type **Double**.

Cliquez sur **OK** pour fermer l’éditeur du schéma. Dans la boîte de dialogue qui s’ouvre, cliquez sur **Yes** pour propager le schéma au composant suivant.

7. Appuyez sur les touches **Ctrl+S** pour enregistrer le Job.
8. Appuyez sur **F6** pour exécuter le Job.

Comme affiché ci-dessus, les informations sur les livres sont extraites des données source JSON et affichées sous forme de table plate dans la console.

**Scénario 3 : Extraire des données JSON d’un fichier en utilisant XPath**

Basé sur **Scénario 1 : Extraire des données JSON d’un fichier en utilisant le JSONPath sans configurer de nœud de boucle** à la page 1088, ce scénario montre comment extraire le nom du magasin et les informations sur les livres à partir du fichier JSON *Store.json* qui utilise des requêtes XPath et comment afficher les données plates extraites dans la console.

**Procédure**

1. Dans le Studio, ouvrez le Job utilisé dans **Scénario 1 : Extraire des données JSON d’un fichier en utilisant le JSONPath sans configurer de nœud de boucle** à la page 1088 pour l’afficher dans l’espace de modélisation graphique.
2. Double-cliquez sur le composant **tFileInputJSON** pour ouvrir sa vue **Basic settings**.
3. Sélectionnez Xpath dans la liste déroulante Read By.

4. Cliquez sur le bouton [...] à côté de Edit schema pour ouvrir l'éditeur du schéma.

Sélectionnez les cinq colonnes ajoutées précédemment et cliquez sur le bouton [x] pour les supprimer toutes.


Cliquez sur OK pour fermer l'éditeur du schéma. Dans la boîte de dialogue qui s'ouvre, cliquez sur Yes pour propager le schéma au composant suivant.

5. Dans le champ Loop XPath query, saisissez la requête XPath entre guillemets doubles pour spécifier sur quel nœud la boucle doit se baser. Dans cet exemple, "/store/goods/book".

6. Dans les champs XPath query de la table Mapping, saisissez les requêtes XPath entre guillemets doubles pour spécifier les nœuds contenant les données souhaitées.

   • Pour la colonne store_name, saisissez la requête XPath ".//name" relative au nœud name.


7. Appuyez sur les touches Ctrl+S pour enregistrer le Job.

8. Appuyez sur F6 pour exécuter le Job.

Comme affiché ci-dessus, le nom du magasin, son adresse et les informations sur les livres sont extraites des données source JSON et affichées sous forme de table plate dans la console.
Scénario 4 : Extraire des données JSON d’une URL

Dans ce scénario, le tFileInputJSON récupère des données du nœud friends du fichier JSON facebook.json sur le Web. Ce fichier contient les données d’un utilisateur de Facebook que le tExtractJSONFields extrait à partir du nœud friends pour les écrire en données plates.

Le fichier JSON facebook.json est déployé sur le serveur Tomcat, situé dans le dossier <tomcat path>/webapps/docs. Le contenu du fichier se présente comme suit :

```json
{"user": {
   "id": "9999912398",
   "name": "Kelly Clarkson",
   "friends": [
      {
         "name": "Tom Cruise",
         "id": "55555555555555",
         "likes": {
            "data": [
               {
                  "category": "Movie",
                  "name": "The Shawshank Redemption",
                  "id": "103636093053996",
                  "created_time": "2012-11-20T15:52:07+0000"
               },
               {
                  "category": "Community",
                  "name": "Positiveretribution",
                  "id": "471389562899413",
                  "created_time": "2012-12-16T21:13:26+0000"
               }
            ]
         }
      },
      {
         "name": "Tom Hanks",
         "id": "88888888888888",
         "likes": {
            "data": [
               {
                  "category": "Journalist",
                  "name": "Janelle Wang",
                  "id": "136009823148851",
                  "created_time": "2013-01-01T08:22:17+0000"
               },
               {
                  "category": "Tv show",
                  "name": "Now With Alex Wagner",
                  "id": "305948749433410",
                  "created_time": "2012-11-20T06:14:10+0000"
               }
            ]
         }
      }
   ]
}}
```
Ajouter et relier les composants

Procédure

1. Créez un nouveau Job et ajoutez un composant tFileInputJSON, un tExtractJSONFields et deux tLogRow en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le composant tFileInputJSON au premier tLogRow à l’aide d’un lien Row > Main.
3. Reliez le premier tLogRow au tExtractJSONFields à l’aide d’un lien Row > Main.
4. Reliez le composant tExtractJSONFields au second tLogRow à l’aide d’un lien Row > Main.

Configurer les composants

Procédure

1. Double-cliquez sur le composant tFileInputJSON pour ouvrir sa vue Basic settings.

   ![Basic settings](image)

   3. Cliquez sur le bouton [...] à côté de Edit schema et, dans la boîte de dialogue [Schema], définissez le schéma en ajoutant une colonne friends de type String.

   ![Schema](image)

   Cliquez sur OK pour fermer la boîte de dialogue et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.
4. Dans la table Mapping, saisissez la requête JSONPath "$user.friends[*]" à côté de la colonne friends pour récupérer le nœud friends entier à partir du fichier source.

5. Double-cliquez sur le tExtractJSONFields pour ouvrir sa vue Basic settings.

6. Sélectionnez Xpath dans la liste déroulante Read By.

7. Dans le champ Loop XPath query, saisissez l’expression XPath entre guillemets doubles pour spécifier le nœud sur lequel la boucle doit se baser. Dans cet exemple, il s’agit de "/likes/data".

8. Cliquez sur le bouton [...] à côté de Edit schema et, dans la boîte de dialogue [Schema], définissez le schéma en ajoutant cinq colonnes de type String, id, name, like_id, like_name et like_category, qui vont contenir les données des nœuds associés sous le champ JSON friends.

Cliquez sur OK pour fermer la boîte de dialogue et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

9. Dans les champs XPath query de la table Mapping, saisissez les requêtes XPath entre guillemets doubles pour spécifier les nœuds JSON qui doivent contenir les données souhaitées. Dans cet exemple:

- ".//id" (effectue une requête sur le nœud "/friends/id") pour la colonne id,
- ".//name" (effectue une requête sur le nœud "/friends/name") pour la colonne name,
- "id" pour la colonne like_id,
• "name" pour la colonne like_name et
• "category" pour la colonne like_category.

10. Double-cliquez sur le second composant tLogRow pour ouvrir sa vue Basic settings.

Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour afficher un résultat plus lisible.

Exécuter le Job

Procédure
1. Appuyez sur les touches Ctrl + S pour enregistrer le Job.
2. Appuyez sur F6 pour exécuter le Job.

```
[statistics] connecting to socket on port 2401
[statistics] connected
  name" "Tom Cruise"" id" #555555555555555555" "like_id" "666666666666666666" "like_name" "The
game of Thrones" ",id" #11111111111111111111" "like_id" "77777777777777777777" "like_name" "Harry
Potter"

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>like_id</th>
<th>like_name</th>
<th>like_category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tom Cruise</td>
<td>12345678901234567890</td>
<td>The Game of Thrones</td>
<td>Movie</td>
</tr>
<tr>
<td>2</td>
<td>Jannick Waeg</td>
<td>12345678901234567890</td>
<td>Jannick Waeg</td>
<td>Journalist</td>
</tr>
</tbody>
</table>

[statistics] disconnected
```

Comme affiché ci-dessus, les données sur les amis (friends) dans le fichier JSON spécifié en utilisant l’URL sont extraites puis les données du nœud friends sont extraites et affichées dans une table plate.
tFileInputLDIF

Ce composant lit un fichier LDIF ligne par ligne afin de le diviser en champs et envoie les champs comme défini, au composant suivant, via une connexion de type Row.

Propriétés du tFileInputLDIF Standard

Ces propriétés sont utilisées pour configurer le tFileInputLDIF s’exécutant dans le framework de Jobs Standard.

Le composant tFileInputLDIF Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>add operation as prefix when the entry is modify type</td>
<td>Cochez cette case pour afficher le mode d’opérations effectuées.</td>
</tr>
<tr>
<td>Value separator</td>
<td>Saisissez le séparateur requis pour découper les données dans un fichier donné. Par défaut, le séparateur utilisé est &quot;,&quot;,.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Reject.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
</tbody>
</table>
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Advanced settings

**Encoding**

Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

**Remarque :**

Le type de données des colonnes qui sera traité par cette option est **byte[]** et vous devrez la définir dans l’éditeur du schéma d’entrée.

**Use field options (for Base64 decode checked)**

Cochez cette case afin de spécifier les colonnes encodées en Base64 dans le flux d’entrée. Une fois cochée, cette case active la table **Decode Base64 encoding values**, vous permettant ainsi de préciser les colonnes à décoder de Base64.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant.

### Global Variables

**Global Variables**

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaine de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.
### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Utilisez ce composant pour lire des lignes complètes dans un fichier délimité pouvant être très gros. Ce composant permet de créer un flux de données à l’aide d’un lien <strong>Row &gt; Main</strong>, ainsi que de créer un flux de rejet avec un lien <strong>Row &gt; Reject</strong> filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d’utilisation de ces deux liens, consultez <em>Procédure</em> à la page 1152 du composant <strong>tFileInputXML</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé, consultez *Scénario : Écrire des données d’une table d’une base de données dans un fichier LDIF* à la page 1191.
tFileInputMail

Ce composant lit les données clés standard d’un fichier e-mail MIME ou MSG défini.

**Propriétés du tFileInputMail Standard**

Ces propriétés sont utilisées pour configurer le tFileInputMail s’exécutant dans le framework de Jobs Standard.

Le composant tFileInputMail Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>File Name</th>
<th>Spécifiez le fichier e-mail dont vous souhaitez lire et extraire les données.</th>
</tr>
</thead>
</table>
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local *(Built-in)*, soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
  - **View schema** : sélectionnez cette option afin de voir le schéma.
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| **Built-in** | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| **Repository** | Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le Guide utilisateur du Studio Talend. |
| **Mail type** | Sélectionnez un type d’email dans la liste déroulante : MIME ou MSG. |
| **Attachment export directory** | Spécifiez le répertoire dans lequel vous souhaitez exporter les pièces jointes. |
### Mail parts

Spécifiez pour chaque colonne de schéma les données à extraire du fichier e-mail MIME défini.

- **Column**: Les cellules **Column** sont automatiquement renseignées avec le nom de colonne défini dans le schéma que vous avez propagé.

- **Mail part**: Saisissez l’étiquette d’en-tête ou de corps de message à extraire de l’e-mail défini.

- **Multi value**: Cochez cette case pour chaque colonne comportant des champs à valeurs multiples.

- **Separator**: Saisissez le séparateur de valeur des champs à valeurs multiples.

Cette table apparaît uniquement lorsque **MIME** est sélectionné dans la liste déroulante **Mail type**.

### MSG Mail parts

Spécifiez pour chaque colonne de schéma les données à extraire du fichier e-mail MSG défini.

- **Column**: Les cellules **Column** sont automatiquement renseignées avec le nom de colonne défini dans le schéma que vous avez propagé.

- **Mail part**: Cliquez sur chaque cellule puis sélectionnez une partie d’e-mail à extraire.

Cette table apparaît uniquement lorsque **MSG** est sélectionné dans la liste déroulante **Mail type**.

### Die on error

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. Décochez la case pour la ligne en erreur et terminer le processus avec les lignes sans erreur.

### Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant.

### Global Variables

**Global Variables**

- **EXPORTED_FILE_PATH**: répertoire dans lequel exporter la pièce jointe de l’e-mail. Cette variable est une variable **Flow** et retourne une chaîne de caractères.

- **ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette...
listé, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

**Règle d’utilisation**

Ce composant traite un flux de données et par conséquent requiert un composant de sortie.

**Scénario : Extraire des champs clés d’un e-mail**

Ce scénario est constitué de deux composants et a pour objectif d’extraire certains champs clés et d’en afficher les valeurs dans la console Run.

![](image)

**Procédure**

1. Cliquez et déposez les composants *tFileInputMail* et *tLogRow*.
2. Reliez les composants à l’aide d’un lien de type *Row Main*.
3. Double-cliquez sur le composant *tFileInputMail* pour afficher l’onglet *Basic settings* de la vue *Component* et configurer les paramètres de base :

   ![](image)

4. Dans le champ *File Name*, sélectionnez le fichier e-mail à traiter. Renseignez manuellement le chemin d’accès ou cliquez sur le bouton [...] pour parcourir vos dossiers jusqu’au fichier choisi.
5. Cliquez sur le bouton [...] à côté du champ *Edit schema* pour ouvrir la boîte de dialogue permettant de définir le schéma à afficher en sortie.
6. Définissez votre schéma en incluant toutes les colonnes que vous souhaitez retrouver dans le flux de sortie. Dans cet exemple, le schéma comporte quatre colonnes *Date, Author, Object* et *Status*.

7. Une fois défini, cliquez sur **OK** pour fermer la boîte de dialogue et propager le schéma dans le tableau *Mail parts*.

8. Dans la colonne *Mail part* du tableau, saisissez les clés standards des parties header (en -tête) et body (corps de message) qui seront utilisées pour retrouver les valeurs à envoyer en sortie.

9. Cochez la case *Multi Value* pour chacune des clés standards pouvant se retrouver plus d'une fois dans votre fichier d'entrée. Définissez si besoin un séparateur pour les valeurs de ces clés dans le champ *Separator*.

10. Double-cliquez sur le composant **tLogRow** pour afficher l'onglet **Basic settings** de la vue **Component** et configurer les paramètres de base.

11. Définissez le composant **tLogRow** afin que les valeurs affichées soient séparées par un retour chariot. Sous Windows, saisissez \\n entre guillemets doubles. Pour plus d'informations sur le composant **tLogRow**, consultez **tLogRow** à la page 2105.

12. Appuyez sur Ctrl+S pour sauvegarder votre Job.

13. Appuyez sur F6 ou sur cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter et afficher le flux de sortie dans la console de la vue **Run Job**.

Starting job **tFileInputMail** at 11:48 06/11/2009.

[statistics] connecting to socket on port 3147
[statistics] connected

Wed, 4 Nov 2009 19:12:47 +0800
(CST) | musicatcher@gmail.com | Talend multi value test | by
10.142.186.14 with SMTP id j14cs69293wff.

Wed, 4 Nov 2009 03:13:00 -0800 (PST) by 10.150.45.40
with SMTP id s4umr2413104bps.250.1257333179981;

Wed, 04 Nov 2009 03:12:59 -0800 (PST) from
mail-gx0-f210.google.com (mail-gx0-f210.google.com
[209.85.217.210])

by nx.google.com with ESMTP id
33e1693716wvh 127.200.11.04.03.12.58;

Wed, 04 Nov 2009 03:12:50 -0800 (PST) by
mail-gx0-f210.google.com with SMTP id 2so6057682gxk.4
for <musicatcher@gmail.com>; Wed, 04 Nov 2009 03:12:58
-0800 (PST) by 10.150.75.12 with SMTP id
x12mr237026yba.341.1257333175484;

Wed, 04 Nov 2009 03:12:55 -0800 (PST) from nsun
([219.237.242.224])

by nx.google.com with ESMTPS id
5em364705ywd.23.200.11.04.03.12.53
(version=SSLv3 cipher-RC4-MD5);

Wed, 04 Nov 2009 03:12:54 -0800 (PST)
[statistics] disconnected

Job **tFileInputMail** ended at 11:48 06/11/2009. [exit code=0]

Résultats

Les valeurs clés d’en-tête sont extraites comme spécifié dans le tableau *Mail parts*. En effet, la date de réception, l’auteur, le sujet ainsi que les différents statuts contenus dans l’e-mail sont affichés.
tFileInputMSDelimited

Ce composant lit les structures (schémas) de données d’un fichier délimité multi-structure et envoie les champs, comme défini dans les différents schémas aux composants suivants, à l’aide de liens Row.

Propriétés du tFileInputMSDelimited Standard

Ces propriétés sont utilisées pour configurer le tFileInputMSDelimited s’exécutant dans le framework de Jobs Standard.

Le composant tFileInputMSDelimited Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Multi Schema Editor</th>
<th>L’éditeur de schémas [Multi Schema Editor] permet de construire et de configurer le flux de données dans un fichier délimité multi-structure pour pouvoir associer un schéma pour chaque sortie. Pour plus d’informations, consultez L’éditeur de schémas Multi Schema Editor à la page 1107.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Les schémas définis dans l’éditeur de schémas [Multi Schema Editor], ainsi que les types d’enregistrements correspondants, s’affichent automatiquement dans le tableau Output.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Trim all columns</th>
<th>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validate date</td>
<td>Cochez cette case pour vérifier strictement le format de la date par rapport au schéma d’entrée.</td>
</tr>
<tr>
<td>Advanced separator (for numbers)</td>
<td>Cochez cette option pour modifier les séparateurs utilisés pour les nombres : Thousands separator : définissez le séparateur utilisé pour les milliers. Decimal separator : définissez le séparateur utilisé pour les décimaux.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>
Global Variables

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable *After* et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

Utilisation

**Règle d’utilisation**

Utilisez ce composant pour lire des fichiers délimités multi structure et définissez un séparateur de champs pour différencier les champs contenus dans ces fichiers.

**Limitation**


L’éditeur de schémas Multi Schema Editor

L’éditeur [Multi Schema Editor](#) vous permet de définir :

- le chemin d’accès au fichier source,
- les propriétés du fichier source,
- la structure des données pour chacun des schémas de sortie.

Lorsque vous définissez la structure de chaque schéma de sortie dans le [Multi Schema Editor](#), le nom des colonnes des différentes structures s’affichent automatiquement dans la liste du schéma d’entrée des composants qui suivent le tFileInputMSDelimited. Vous pouvez cependant définir la structure des données directement dans la vue *Basic settings* de chacun de ces composants.
Le [Multi Schema Editor] permet également de sélectionner un schéma qui servira de source (clé primaire) aux données entrantes, afin d’en assurer l’unicité. L’éditeur utilise ce procédé de mapping pour associer tous les schémas traités dans le fichier délimité au schéma source de ce même fichier.

L’éditeur s’ouvre en affichant par défaut la première colonne. Cette colonne contient généralement l’indicateur de type des enregistrements. Une fois ouvert, vous pouvez cependant définir comme clé primaire n’importe quelle colonne du schéma en cochant la case associée.

L’image ci-dessous donne un aperçu de l’éditeur de schémas [Multi Schema Editor].

Pour des informations plus détaillées sur le Multi Schema Editor, consultez Scénario : Lecture d’un fichier délimité multi-structure à la page 1109.
Scénario : Lecture d'un fichier délimité multi-structure

Le présent scénario sert à créer un Job en Java permettant de lire trois schémas dans un même fichier délimité et d'en afficher la structure sur la console de la vue Run.

Le fichier délimité traité dans ce scénario se présente comme suit :

| 01 | SOFT MUSIC ALBUM; RICHARDSON; 15/12/2005 |
| 02 | We Danced                                |
| 02 | She's Everything                        |
| 02 | Once in a Lifetime Love                 |
| 03 | National Library                         |
| 01 | COUNTRY MUSIC ALBUM; WHITE; 02/01/2006   |
| 02 | Fall Into Me                             |
| 02 | Another Try                              |
| 02 | Something About Her                      |

Déposer et relier les composants

Procédure

1. A partir de la Palette, déposez les composants suivants dans l'espace de modélisation graphique : un composant tFileInputMSDelimited et trois tLogRow.

2. Dans l'espace de modélisation graphique, faites un clic-droit sur le composant tFileInputMSDelimited et connectez-le aux trois composants tLogRow (nommés ici tLogRow1, tLogRow2 et tLogRow3) à l'aide de liens row nommés respectivement row_A_1, row_B_1 et row_C_1.

Configurer les composants

Procédure

1. Double-cliquez sur le tFileInputMSDelimited pour ouvrir l'éditeur Multi Schema Editor.
2. Cliquez sur le bouton **Browse...** à côté du champ **File name** pour indiquer où se trouve le fichier délimité multi-schéma que vous voulez traiter.

3. Dans la zone **File Settings**, renseignez :
   - le type d'encodage de votre fichier source, à partir de la liste déroulante. Cette option permet de garantir une certaine cohérence entre tous les fichiers d'entrée et de sortie.
   - les séparateurs de champ et de ligne utilisés dans le fichier source.

   **Remarque :**
   - Cochez la case **Use Multiple Separator** et configurez les champs qui suivent si différents séparateurs de champs sont utilisés pour séparer les schémas du fichier source.

   Un aperçu des données du fichier source apparaît automatiquement dans le panneau d'affichage **Preview**.
Remarque : La colonne Column 0, qui contient généralement l’indicateur de type des enregistrements est sélectionnée par défaut. Vous pouvez cependant définir n’importe quelle autre colonne comme clé primaire en cochant sa case associée.

4. Cliquez sur le bouton Fetch Codes à droite du panneau Preview pour cataloguer les types de schémas et les enregistrements du fichier source. Dans ce scénario, le fichier source comporte trois types de schémas (A, B, C).

A partir du panneau d’affichage Fetch Codes, cliquez sur chacun des trois types pour que la structure de leurs données s’affiche dans le panneau Preview.

5. Cliquez dans les cellules de la ligne Name et choisissez le nom des colonnes de chacun des schémas sélectionnés.

Dans ce scénario, les colonnes se lisent comme suit :

-Schéma A : Type, DiscName, Author, Date,
- Schéma B : Type, SongName,
- Schéma C : Type, LibraryName.

Vous devez à présent définir, à partir des données d’entrée, la clé primaire qui permettra de conserver leur unicité (dans ce scénario, il s’agit de DiscName). Il vous faut pour cela :

6. Sélectionner, à partir du panneau d’affichage Fetch Codes, le schéma contenant la colonne que vous voulez définir comme clé primaire (dans ce scénario, il s’agit du schéma A) pour en afficher la structure.

7. Cliquer dans la cellule Key correspondant à la colonne DiscName et cocher la case qui s’affiche alors.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>DiscName</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>TagLevel</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
<td>String</td>
<td>String</td>
<td>String</td>
</tr>
<tr>
<td>Length</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Pattern</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. Cliquer n’importe où dans l’éditeur pour que la mention false, dans la case Key, devienne true.

Vous devez maintenant indiquer le schéma “père” auquel vous voulez rattacher les autres schémas “fils” (dans ce scénario, il s’agit de DiscName). Il vous faut pour cela :

9. Sélectionner le schéma B, à partir du panneau d’affichage Fetch Codes, puis cliquer sur le bouton flèche droite pour le déplacer vers la droite.

Faire de même pour le schéma C.

Remarque :

Le champ Cardinality n’est pas obligatoire. Il vous permet de déterminer le nombre (ou l’intervalle) de champs des schémas “fils” rattachés au schéma “père”. Cependant, si vous entrez un nombre ou un intervalle inexact et essayez d’exécuter le Job, un message d’erreur apparaîtra.

10. Dans le [Multi Schema Editor], cliquez sur le bouton OK pour valider tous les changements que vous venez d’effectuer et fermer l’éditeur de schémas.

Les trois schémas définis, ainsi que les types d’enregistrements correspondants, s’affichent automatiquement dans la vue Basic settings du composant tFileInputMSDelimited.
Les trois schémas que vous avez définis dans l’éditeur [Multi Schema Editor] sont automatiquement communiqués aux trois composants tLogRow.

11. Si nécessaire, cliquez sur le bouton Edit schema dans la vue Basic settings de chacun des composants tLogRow pour avoir un aperçu des structures de données en entrée et en sortie, telles que vous les avez définies dans le Multi Schema Editor, ou pour les modifier.

Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur Ctrl+S pour sauvegarder votre Job.
2. Appuyez sur F6 ou cliquez sur le bouton Run de l’onglet Run pour l’exécuter.

Le fichier délimité multi-schéma est lu ligne par ligne et les champs extraits sont affichés dans la console Run Job comme défini dans le [Multi Schema Editor].
Starting job pivot at 10:33 18/01/2010.

<table>
<thead>
<tr>
<th></th>
<th>TRACK</th>
<th>ARTIST</th>
<th>ALBUM</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFT MUSIC ALBUM</td>
<td>RICHARDSON</td>
<td>15/12/2005</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>We Danced</td>
<td>SOFT MUSIC ALBUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>She’s Everything</td>
<td>SOFT MUSIC ALBUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Once in a Lifetime Love</td>
<td>SOFT MUSIC ALBUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>National Library</td>
<td>SOFT MUSIC ALBUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>COUNTRY MUSIC ALBUM</td>
<td>WHITE</td>
<td>02/01/2006</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Fall Into Me</td>
<td>COUNTRY MUSIC ALBUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Another Try</td>
<td>COUNTRY MUSIC ALBUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Something About Her</td>
<td>COUNTRY MUSIC ALBUM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Job pivot ended at 10:33 18/01/2010. [exit code=0]
tFileInputMSPositional

Ce composant lit les structures de données (schémas) d’un fichier positionnel multi-structure et envoie les champs, comme défini dans les différents schémas, aux composants suivants, à l’aide de liens Row.

Propriétés du tFileInputMSPositional Standard

Ces propriétés sont utilisées pour configurer le tFileInputMSPositional s’exécutant dans le framework de Jobs Standard.

Le composant tFileInputMSPositional Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>File Name</strong></td>
<td>Chemin d’accès et nom du fichier, et/ou variable à traiter. Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Row separator</strong></td>
<td>Chaîne de caractères (ex: &quot;\n&quot; sous Unix) séparant les lignes.</td>
</tr>
<tr>
<td><strong>Header Field Position</strong></td>
<td>Position de la chaîne de caractères identifiant les différents schémas.</td>
</tr>
<tr>
<td><strong>Records</strong></td>
<td><strong>Schema</strong> : définissez autant de schémas que nécessaire. <strong>Header value</strong> : valeur de la chaîne de caractères identifiant les différents schémas. <strong>Pattern</strong> : Chaîne de caractères représentant la longueur de chacune des colonnes du schéma correspondant, séparées par des virgules. Vérifiez que les valeurs définies dans ce champ sont cohérentes avec le schéma défini. <strong>Reject incorrect row size</strong> : Cochez les cases des schémas pour lesquels vous n’acceptez pas une taille de ligne incorrecte. <strong>Parent row</strong> : Sélectionnez la ligne parent dans la liste déroulante. Par défaut, il s’agit d’&lt;Empty&gt;.</td>
</tr>
</tbody>
</table>
### Parent key column
Saisissez le nom de la colonne clé parente. Si la ligne parent n’est pas «Empty», vous devez saisir le nom de la colonne de la ligne parent du schéma dans ce champ.

### Key column
Saisissez le nom de la colonne clé.

### Skip from header
Nombre de lignes à ignorer au début du fichier.

### Skip from footer
Nombre de lignes à ignorer à la fin du fichier.

### Limit
Nombre maximal de lignes à traiter. Si Limit = 0, aucune ligne ne sera lue ni traitée.

### Die on parse error
Décochez cette case pour passer les lignes contenant des erreurs de parse et terminer le traitement avec les lignes sans erreur.

### Die on unknown header type
Décochez cette case pour passer les lignes dont le type d’en-tête est inconnu et terminer le traitement avec les lignes sans erreurs.

### Advanced settings

<table>
<thead>
<tr>
<th>Process long rows (needed for processing rows longer than 100,000 characters wide)</th>
<th>Cochez cette case afin de pouvoir traiter de longues lignes (ceci est nécessaire pour traiter des lignes contenant plus de 100 000 caractères).</th>
</tr>
</thead>
</table>
| Advanced separator (for numbers) | Cochez cette option pour modifier les séparateurs utilisés pour les nombres :

- **Thousands separator** : définissez le séparateur utilisé pour les milliers.
- **Decimal separator** : définissez le séparateur utilisé pour les décimaux. |
| Trim all columns | Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes. |
| Validate date | Cochez cette case pour vérifier strictement le format de la date par rapport au schéma d’entrée. |
| Encoding | Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données. |
| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Global Variables

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **Flow** et retourne un entier. |
NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable Flow et retourne un nombre entier.

NB_LINE_UNKNOWN_HEADER_TYPES : nombre de lignes ayant un type d’en-tête inconnu. Cette variable est une variable Flow et retourne un nombre entier.

NB_LINE_PARSE_ERRORS : nombre de lignes ayant des erreurs de parsage. Cette variable est une variable Flow et retourne un nombre entier.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | Utilisez ce composant pour lire un fichier positionnel multi-schéma et servez-vous d’une valeur spécifiant l’emplacement pour séparer les champs. Ce composant permet de créer un flux de données à l’aide d’un lien Row > Main, ainsi que de créer un flux de rejet avec un lien Row > Reject filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d’utilisation de ces deux liens, consultez Procédure à la page 1152 du composant tFileInputXML. |

### Scénario : Lire des données d’un fichier positionnel

Le scénario suivant permet de lire des données d’un fichier positionnel contenant deux schémas. Le fichier positionnel se présente comme suit :

```
schema_1 (car_owner):schema_id;car_make;owner;age
1bmw John  45
1bench Mike  30
2John  45 yes
2Mike  50 No

schema_2 (car-insurance):schema_id;car_owner;age;car_insurance
```
Déposer les composants

Procédure

1. Déposez un tFileInputMSPositional et deux composants tLogRow de la Palette dans l’espace de modélisation graphique.

2. Renommez les deux composants tLogRow, respectivement car_owner et car_insurance.

Configurer les composants

Procédure

1. Double-cliquez sur le tFileInputMSPositional pour afficher sa vue Basic settings et paramétrer ses propriétés.

2. Dans le champ File name/Stream, saisissez le chemin d’accès à votre fichier d’entrée. Vous pouvez également cliquer sur le bouton [...] pour parcourir votre système et sélectionner votre fichier.

3. Dans le champ Header Field Position, saisissez la position de début et de fin de l’en-tête pour l’identifiant du schéma dans le fichier d’entrée, 0-1, dans ce cas, puisque le premier caractère de chaque ligne est l’identifiant du schéma.
4. Cliquez deux fois sur le bouton [+] pour ajouter deux lignes à la table Records.

5. Cliquez sur la cellule dans la colonne Schema pour afficher le bouton [...].
   Cliquez sur le bouton [...] pour ouvrir la boîte de dialogue de nommage du schéma.

![Give the name for the schema dialog box](image1)

6. Saisissez le nom du schéma et cliquez sur OK.
   Le nom du schéma apparaît dans la cellule et l’éditeur du schéma s’ouvre.

![tFileInputMSPositional_3](image2)

8. Répétez ces étapes pour définir le schéma car_insurance, contenant quatre colonnes : schema_id, car_owner, age et car_insurance.

![tFileInputMSPositional_3](image3)

10. Dans la colonne Header value, saisissez l’identifiant du schéma, 1 pour le schéma car_owner et 2 pour le schéma car_insurance.
11. Dans la colonne **Pattern**, saisissez la longueur de chaque champ du schéma, c'est-à-dire le nombre de caractères, 1,8,10,3 pour le schéma `car_owner` et 1,10,3,3 pour le schéma `car_insurance` dans ce scénario.

12. Dans le champ **skip from header**, saisissez le nombre de lignes d'en-tête à ignorer, 2 dans cet exemple, car les deux premières lignes décrivent les deux schémas et ne fournissent aucune valeur.

13. Sélectionnez **Table (print values in cells of a table)** dans la zone **Mode** des composants `car_owner` et `car_insurance`.

**Enregistrer et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour exécuter le Job.

```
La procédure est exécutée.
```

Le fichier est lu ligne par ligne selon les valeurs de longueur définies dans le champ **Pattern** et écrit en sortie dans deux tables ayant deux schémas différents.
Ce composant lit les structures de données (schémas) d’un fichier XML multi-structure et envoie les champs, comme défini dans les différents schémas, aux composant suivants, à l’aide de liens Row.

**Propriétés du tFileInputMSXML Standard**

Ces propriétés sont utilisées pour configurer le tFileInputMSXML s’exécutant dans le framework de Jobs Standard.

Le composant tFileInputMSXML Standard appartient aux familles File et XML.

Le composant de ce framework est toujours disponible.

### Basic settings

| File Name | Chemin d’accès et nom du fichier, et/ou variable à traiter.  
Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Root XPath query</td>
<td>Racine de l’arborescence XML sur laquelle se base la requête.</td>
</tr>
</tbody>
</table>
| Enable XPath in column “Schema XPath loop” but lose the order | Cochez cette case si vous souhaitez définir un chemin XPath dans le champ Schema XPath loop du tableau Outputs sans garder l’ordre des données montrées dans le fichier XML source.  
**Avertissement :**  
Cette option ne prend effet que si vous sélectionnez le mode de génération dom4j dans la vue Advanced settings. |
| Outputs |  
**Schema** : Définissez autant de schémas que nécessaire.  
**Schema XPath loop** : Saisissez le noeud ou le chemin XPath de l’arborescence XML sur lequel la boucle est basée.  
**XPath Queries** : Renseignez les champs à extraire de la structure XML d’entrée.  
**Create empty row** : Cochez cette case si vous souhaitez créer des lignes vides pour le(s) champ(s) vide(s) du schéma. |
| Die on error | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. |
## Advanced settings

<table>
<thead>
<tr>
<th><strong>Trim all column</strong></th>
<th>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Validate date</strong></td>
<td>Cochez cette case pour vérifier strictement le format de la date par rapport au schéma d'entrée.</td>
</tr>
<tr>
<td><strong>Ignore DTD file</strong></td>
<td>Cochez cette case afin d’ignorer le fichier DTD indiqué dans le fichier XML traité.</td>
</tr>
<tr>
<td><strong>Generation mode</strong></td>
<td>Sélectionnez le mode de génération correspondant à votre mémoire disponible. Les modes disponibles sont :</td>
</tr>
<tr>
<td></td>
<td>• Lent et consommateur de mémoire (<em>Slow and memory-consuming</em> - Dom4J).</td>
</tr>
<tr>
<td></td>
<td><em>Remarque :</em></td>
</tr>
<tr>
<td></td>
<td>Cette option vous permet d'utiliser Dom4J pour traiter des fichiers XML très complexes.</td>
</tr>
<tr>
<td></td>
<td>• Rapide et peu consommateur de mémoire (<em>Fast with low memory consumption</em> (SAX)).</td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

## Global Variables

| **Global Variables** | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable After et retourne un entier. |
|                      | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. |
|                      | Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |
|                      | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
|                      | Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |
Scénario : Lecture d'un fichier XML multi-structuré

Le présent scénario décrit un Job permettant de lire un fichier XML multi-structuré, d'en extraire les champs désirés et de les afficher dans la console.

Construire le Job

Procédure

1. Déposez un tFileInputMSXML de la Palette dans l'espace de modélisation graphique et double-cliquez sur le composant afin d'ouvrir sa vue Basic settings dans l'onglet Component.

2. Parcourez votre système jusqu'au fichier XML que vous souhaitez traiter, D:/Input/multi schema_xml.xml dans cet exemple. Ce fichier qui contient les données suivantes :

```xml
<root>
  <toy>Cat</toy>
  <record>We Belong Together</record>
  <book>As You Like It</book>
  <book>All's Well That Ends Well</book>
  <record>When You Believe</record>
  <toy>Dog</toy>
</root>
```


5. Cliquez sur le bouton [+] pour ajouter des lignes au tableau Outputs dans lesquelles vous pouvez définir les schémas de sortie, record et book dans cet exemple.

6. Dans le tableau Outputs, cliquez dans chaque cellule Schema et cliquez sur le bouton [...] pour saisir le nom du schéma dans la boîte de dialogue qui s’ouvre.

Saisissez un nom pour le schéma de sortie et cliquez sur OK afin de fermer la boîte de dialogue.
7. L’éditeur de schéma du tFileInputMSXML s’affiche.
Définissez le schéma selon vos besoins.

8. Effectuez cette procédure afin de définir le schéma de sortie record.


10. Dans la cellule XPath Queries, indiquez les champs à extraire du fichier XML d’entrée. Dans cet exemple, saisissez la requête XPath ".".

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** ou cliquez sur **Run** dans l'onglet **Run** afin d'exécuter le Job.

Le fichier XML multi-structuré est lu ligne par ligne et les champs extraits s’affichent dans la console. Les deux premiers champs sont pour le schéma *book* et les deux autres pour le schéma *record*.

**Starting job check at 17:58 07/05/2014.**

[statistics] connecting to socket on port 3985
[statistics] connected
As You Like It
All's Well That Ends Well
We Belong Together
When You Believe
[statistics] disconnected
Job check ended at 17:58 07/05/2014. [exit code]
tFileInputPositional

Ce composant lit un fichier positionnel ligne par ligne afin de séparer les champs en se basant sur un modèle donné. Il envoie ensuite les champs, comme défini dans le schéma, au composant suivant.

Propriétés du tFileInputPositional Standard

Ces propriétés sont utilisées pour configurer le tFileInputPositional s’exécutant dans le framework de Jobs Standard.

Le composant tFileInputPositional Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
</tbody>
</table>

#### File name/Stream

- **File name** : Chemin d’accès et nom du fichier à traiter.
- **Stream** : Flux de données à traiter. Les données doivent préalablement être mises en flux afin d’être récupérées par le tFileInputPositional via la variable représentative correspondante.

Cette variable peut être prédéfinie dans votre Studio ou fournie par le contexte ou les composants utilisés avec ce composant, par exemple la variable `INPUT_STREAM` du tFileFetch. Sinon, vous pouvez la définir manuellement et l’utiliser selon votre Job, par exemple à l’aide d’un tJava ou d’un tJavaFlex.

Afin d’éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d’autocomplétion (Ctrl+Espace) afin de remplir le champ, si cette variable a été correctement définie.

Pour plus d’informations concernant les variables disponibles, consultez le Guide utilisateur du Studio Talend.

Pour plus d’informations concernant les flux d’entrée, consultez Scénario 2 : Lire les données d’un fichier distant en mode stream à la page 1072.

<table>
<thead>
<tr>
<th>Row separator</th>
<th>Saisissez le séparateur utilisé pour identifier la fin d’une ligne.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use byte length as the cardinality</td>
<td>Cochez cette case pour permettre la prise en charge des caractères à deux octets dans ce composant. Pour cette fonction, JDK 1.6 est requis.</td>
</tr>
</tbody>
</table>
| **Customize** | Cochez cette case pour personnaliser le format des données du fichier positionnel et renseignez les colonnes du tableau Formats.  
- **Column** : Sélectionnez la colonne que vous souhaitez personnaliser.  
- **Size** : Saisissez la taille correspondant à la colonne.  
- **Padding char** : saisissez, entre guillemets, le caractère de remplissage à supprimer du champ. Le caractère par défaut est un espace.  
- **Alignment** : Sélectionnez le paramètre d’alignement approprié. |
| **Pattern** | Longueurs séparées par des virgules, interprétées comme une chaîne de caractères entre guillemets. Vérifiez que les valeurs saisies dans ce champ sont cohérentes avec le schéma défini. |
| **Pattern Units** | Unité des valeurs de longueur spécifiées dans le champ Pattern.  
- **Bytes** : lorsque cette option est sélectionnée, les valeurs de longueur dans le champ Pattern doivent être le nombre d’octets représentant les symboles dans l’encodage original du fichier d’entrée.  
- **Symbols** : lorsque cette option est sélectionnée, les valeurs de longueur dans le champ Pattern doivent être le nombre de symboles habituels, sans inclure les paires de substitution.  
- **Symbols (including rare)** : lorsque cette option est sélectionnée, les valeurs de longueur dans le champ Pattern doivent être le nombre de symboles, y compris les symboles inhabituels, comme les paires de substitution. Chaque paire de substitution compte comme un seul symbole. Au niveau des performances, il n’est pas recommandé d’utiliser cette option lorsque vos données d’entrée comprennent uniquement des symboles habituels. |
| **Skip empty rows** | Cochez cette case pour ignorer les lignes vides. |
| **Uncompress as zip file** | Cochez cette case pour décompresser le fichier d’entrée. |
| **Die on error** | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.  
Décchez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject. |
<p>| <strong>Header</strong> | Saisissez le nombre de lignes à ignorer au début du fichier. |
| <strong>Footer</strong> | Nombre de ligne à ignorer à la fin d’un fichier. |</p>
<table>
<thead>
<tr>
<th>Limit</th>
<th>Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n’est lue ou traitée.</th>
</tr>
</thead>
</table>
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• **View schema** : sélectionnez cette option afin de voir le schéma.  
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].  
Pour bénéficier de la fonctionnalité de schéma dynamique, vous pouvez utiliser ce composant avec le **tSetDynamicSchema**. |
| **Advanced settings** | **Needed to process rows longer than 100 000 characters** | Cochez cette case si les lignes à traiter dans le fichier d’entrée font plus de 100 000 caractères de long. |
| **Advanced separator (for numbers)** | Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers est une virgule (,) et le séparateur décimal est un point (.).  
**Thousands separator** : configurez le séparateurs des milliers.  
**Decimal separator** : configurez le séparateur pour les décimaux. |
| **Trim all columns** | Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes. |

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
Check columns to trim

Cochez la case devant le nom de chacune des colonnes dont vous souhaitez supprimer les espaces de début et de fin de champ.

Validate date

Cochez cette case pour vérifier strictement le format de date par rapport au schéma d’entrée.

Encoding

 Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant.

Variables globales

Global Variables

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Utilisez ce composant pour lire un fichier et séparer les champs à l’aide du séparateur spécifié. Ce composant permet de créer un flux de données à l’aide d’un lien Row > Main, ainsi que de créer un flux de rejet avec un lien Row > Reject filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d’utilisation de ces deux liens, consultez Procédure à la page 1152 du composant tFileInputXML.

Transformer un fichier positionnel en fichier XML

Le scénario suivant construit un Job avec deux composants, qui a pour objectif de lire les données d’un fichier positionnel en entrée et de rendre des données sélectionnées en sortie (selon leur position) dans un fichier XML.

Contract CustomerRef InsuranceNr
Déposer et relier les composants

Pourquoi et quand exécuter cette tâche

Procédure

1. Cliquez et déposez un composant tFileInputPositional de la Palette dans l’espace de modélisation.
2. Cliquez-déposez un composant tFileOutputXML. Ce fichier recevra les références de manière structurée.
3. Cliquez-droit sur le composant tFileInputPositional et sélectionnez une connexion Row > Main. Glissez cette connexion vers le composant tFileOutputXML et relâchez la souris lorsque le symbole de prise de courant apparaît.

Configurer les données d’entrée

Procédure

1. Sélectionnez le composant tFileInputPositional pour afficher sa vue Basic settings et définir ses propriétés.
2. Les propriétés de ce Job sont de type Built-in pour ce scénario. Par conséquent, les informations de propriétés sont renseignées pour ce Job seulement et ne peuvent être réutilisées pour un autre Job, contrairement à des propriétés de type Repository.
3. Renseignez le chemin d’accès au fichier dans le champ File Name. Ce champ est obligatoire.
4. Puis définissez le séparateur de lignes (Row separator) permettant d’identifier la fin de la ligne : le retour chariot par défaut.
5. Si nécessaire, cochez la case **Use byte length as the cardinality** pour permettre la prise en charge des caractères à deux octets.

6. Puis dans le champ **Pattern** définissant les champs d’une ligne. Le pattern est une série de longueurs correspondant aux valeurs de champs du fichier en entrée. Les valeurs doivent être saisis entre guillemets simples et séparées par une virgule. Veillez à ce que les valeurs saisis correspondent à la longueur des champs définis dans le schéma.

7. Renseignez les champs d’en-tête (**Header**), de pied de page (**Footer**) et de limite (**Limit**) selon la structure de votre fichier d’entrée et selon vos besoins. Dans ce scénario, ignorez la première ligne lors de la lecture du fichier d’entrée en saisisant 1 dans le champ **Header** et laissez les autres champs tels qu’ils sont.

8. Dans la liste **Schema**, sélectionnez **Repository** si le schéma d’entrée est stocké dans le Repository. Dans ce scénario, sélectionnez **Built-In** pour définir les données à transmettre au composant **tFileOutputXML**.

9. Vous pouvez sélectionner et/ou modifier le schéma via la fonction **Edit Schema**. Pour ce schéma, définissez trois colonnes, respectivement **Contracts**, **CustomerRef** et **InsuranceNr** correspondant aux trois valeurs de longueurs définies. Cliquez sur **OK** pour fermer la boîte de dialogue [Schema] et propager les modifications.

![Schema of Contracts](image)

**Configurer les données de sortie**

**Procédure**

1. Double-cliquez sur le composant **tFileOutputXML** afin d’afficher sa vue **Basic settings** et configurer ses propriétés de base.

![Contracts_Ref_XML](image)

2. Saisissez le chemin d’accès au fichier XML de sortie.
3. Définissez la balise de la ligne (Row tag) qui définit chaque ligne. Dans ce cas, la balise est **ContractRef**.

4. Cliquez sur le bouton [+] à côté du champ **Edit Schema** pour voir la structure de données, puis cliquez sur **Sync columns** pour récupérer la structure des données du composant d’entrée.

5. Cliquez sur l’onglet **Advanced settings** afin de configurer les paramètres avancées de la sortie

<table>
<thead>
<tr>
<th>Split output in several files</th>
<th>Create directory if not exists</th>
</tr>
</thead>
</table>

### Root tags

<table>
<thead>
<tr>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;ContractsList&quot;</td>
</tr>
</tbody>
</table>

### Output format

<table>
<thead>
<tr>
<th>Column</th>
<th>As attribute</th>
<th>Use schema column name</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contract</td>
<td></td>
<td></td>
<td>&quot;label&quot;</td>
</tr>
<tr>
<td>CustomerRef</td>
<td></td>
<td></td>
<td>&quot;label&quot;</td>
</tr>
<tr>
<td>InsuranceNr</td>
<td></td>
<td></td>
<td>&quot;label&quot;</td>
</tr>
</tbody>
</table>

6. Cliquez sur le bouton [+] pour ajouter une ligne dans la table **Root tags** et saisissez un nom de balise (ou plusieurs) pour encadrer la structure XML de sortie, **ContractsList** dans ce scénario.

7. Configurez les paramètres dans la table **Output format** si nécessaire. Par exemple, cochez la case **As attribute** d’une colonne si vous souhaitez utiliser son nom et sa valeur comme attribut pour l’élément XML parent. Décochez la case **Use schema column name** pour que la colonne réutilise le libellé de la colonne d’entrée comme libellé de la balise. Dans ce scénario, laissez les paramètres par défaut.

8. Pour regrouper les lignes de sortie selon le numéro de contrat, cochez la case **Use dynamic grouping**, ajoutez une ligne dans la table **Group by**, sélectionnez **Contract** dans la liste **Column** et saisissez un attribut pour cette colonne dans le champ **Attribute label**.

9. Laissez les autres paramètres tels qu’ils sont.
Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job et vous assurer que tous les paramètres sont bien pris en compte.

2. Appuyez sur **F6** ou cliquez sur l'onglet **Run** puis sur le bouton **Run** pour exécuter le Job.

Le fichier est lu ligne par ligne selon les longueurs définies précédemment dans le champ **Pattern** et écrit en tant que fichier XML, comme défini dans les paramètres de sortie. Vous pouvez l’ouvrir dans n’importe quel éditeur XML standard.

```xml
<xml version="1.0" encoding="ISO-8859-15">
  <ContractsList>
    <Contract Nr="00001">
      <ContractRef>
        <CustomerRef>8200</CustomerRef>
        <InsuranceNr>50330</InsuranceNr>
      </ContractRef>
    </Contract>
    <Contract Nr="00002">
      <ContractRef>
        <CustomerRef>8201</CustomerRef>
        <InsuranceNr>50331</InsuranceNr>
      </ContractRef>
    </Contract>
    <Contract Nr="00003">
      <ContractRef>
        <CustomerRef>8202</CustomerRef>
        <InsuranceNr>50332</InsuranceNr>
      </ContractRef>
    </Contract>
    <Contract Nr="00004">
      <ContractRef>
        <CustomerRef>8203</CustomerRef>
        <InsuranceNr>50333</InsuranceNr>
      </ContractRef>
    </Contract>
  </ContractsList>
</xml>
```
tFileInputProperties

Ce composant lit un fichier texte ligne par ligne, sépare les champs en se basant sur le modèle clé=valeur.

Propriétés du tFileInputProperties Standard

Ces propriétés sont utilisées pour configurer le tFileInputProperties s’exécutant dans le framework de Jobs Standard.

Le composant tFileInputProperties Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être *Built-in* ou distant dans le *Repository*. Le schéma de ce composant est en lecture seule. Il se compose de deux colonnes, *Key* et *Value*, correspondant respectivement au nom du paramètre et à la valeur du paramètre à copier. |
| File format | Sélectionnez le format de votre fichier `.properties` ou `.ini`. |
| `.properties` : fichier de configuration dont les données sont disposées sur deux lignes et structurées de la manière suivante clé=valeur. |
| `.ini` : fichier de configuration dont les données sont disposées sur deux lignes, structurées de la manière suivante clé=valeur et regroupées en sections. |
| **Retrieve Mode** : |
| - **Retrieve All** pour récupérer l’intégralité du fichier, |
| - **Retrieve Sections** pour récupérer les sections du fichier, |
| - **Retrieve by section** pour récupérer les données contenues dans la section spécifiée dans le champ *Section Name*. |
| **Section Name** : saisissez le nom de la section sur laquelle effectuer l’opération d’itération. |
| File | Chemin d’accès et nom du fichier, et/ou variable à traiter. |
| Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le *Guide utilisateur du Studio Talend*. |
Calculate MD5 Hash: Cochez cette case pour que l'algorithme vérifie que le fichier est bien téléchargé.

**Advanced settings**

**Encoding**
Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant.

**Global Variables**

**Global Variables**

**NB_LINE**: nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

**ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

**Règle d’utilisation**
Utilisez ce composant pour lire un fichier et séparer les données en fonction de la structure clé=valeur du fichier.

**Scénario : Lire et mapper la clé et les valeurs de fichiers properties et alimenter un glossaire**

Le scénario suivant est un Job de quatre composants lisant deux fichiers propriétés en anglais et en français. Ces données sont ensuite mises en correspondance afin de constituer un glossaire.

Les fichiers de propriétés utilisés pour ce scénario sont des fichiers contenant les chaînes de caractères utilisés pour la localisation du composant tMysqlInput dans votre Studio Talend.
Le glossaire apparaît dans la console de la vue Run avec dans la première colonne le nom de la clé, dans la deuxième le terme du fichier anglais et dans le troisième colonne le terme correspondant du fichier français.

**Déposer et relier les composants**

**Procédure**

1. Cliquez-glissez deux composants `tFileInputProperties` de la famille File > Input dans le Job designer, un composant `tMap` de la famille Processing et un composant `tLogRow` de la famille Log & Errors.

2. Reliez les composants entre eux via des connexions Row > Main Le second fichier, FR, est utilisé comme flux de référence.

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le premier composant `tFileInputProperties` pour paramétrer ses propriétés dans l’onglet Basic settings de la vue Component.
2. Dans le champ **File Format**, sélectionnez le format de votre fichier.
3. Dans le champ **File Name**, cliquez sur le bouton [...] et sélectionnez le fichier *properties* anglais que vous souhaitez utiliser.
4. Répétez cette procédure avec le deuxième composant et le fichier français :
5. Double-cliquez sur le composant **tMap** pour ouvrir son éditeur :
6. Sélectionnez toutes les colonnes de la table *English_terms* et glissez-les vers la table de sortie.
   Sélectionnez la colonne *key* de la table *English_terms* et glissez-la dans la colonne *key* de la table *French_terms*.

7. Dans le tableau *glossary* de la zone *Schema editor* en bas du *tMap*, renommez le champ *value* en *EN* puisqu’il contiendra les valeurs du fichier anglais.

8. Cliquez sur le bouton [*+] pour ajouter une ligne au tableau *glossary* et renommez-la *FR*.


### Sauvegarder et exécuter le Job

**Procédure**

1. Appuyez sur les touches *Ctrl+S* pour enregistrer votre Job.
2. Appuyez sur *F6* pour l’exécuter ou cliquez sur le bouton *Run* de la vue *Run*.
**tFileInputRaw**

Ce composant lit toutes les données d’un fichier brut et les envoie dans une colonne de sortie pour traitement par un autre composant.

**Propriétés du tFileInputRaw Standard**

Ces propriétés sont utilisées pour configurer le tFileInputRaw s’exécutant dans le framework de Jobs Standard.

Le composant tFileInputRaw Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th><strong>Schema et Edit Schema</strong></th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Filename</strong></td>
<td>Nom et chemin d’accès au fichier d’entrée à traiter. Vous pouvez saisir le chemin manuellement entre guillemets doubles ou parcourir votre système en cliquant sur le bouton [...].</td>
</tr>
<tr>
<td><strong>Mode</strong></td>
<td><strong>Read the file as a string</strong> : Le contenu du fichier est lu comme (string).</td>
</tr>
<tr>
<td></td>
<td><strong>Read the file as a bytes array</strong> : Le contenu du fichier est lu comme bytes array.</td>
</tr>
</tbody>
</table>
Stream the file: Dès que le premier caractère est saisi dans le fichier source, il est immédiatement lu.

**Encoding**
Si vous utilisez le mode *Read the file as a string*, sélectionnez le type d’encodage dans la liste ou sélectionnez *Custom* pour le définir manuellement.

**Die on error**
Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien *Row > Reject*.
Pour capturer l’exception *FileNotFoundException*, vous devez cocher cette case.

**Advanced settings**

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**FILENAME_PATH** : chemin du fichier d’entrée. Cette variable est une variable *After* et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

**Règle d’utilisation**
Ce composant est utilisé pour fournir des données d’entrée pour les Jobs nécessitant une seule colonne de données ou nécessitant qu’un fichier complet soit lu comme une seule colonne.

**Limitation**
Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton *Install* dans l’onglet *Component*. Vous pouvez également trouver les...
Scénario associé

Pour un scénario associé, consultez:

- Scénario : Télécharger des fichiers dans Dropbox à la page 695
tFileInputRegex

Ce composant lit un fichier ligne par ligne afin de le diviser en champs à l’aide d’expressions régulières et d’envoyer les champs définis dans le schéma au composant suivant.

Puissant composant qui peut remplacer bon nombre des composants File. Il requiert des connaissances avancées en rédaction d’expressions régulières.

Propriétés du tFileInputRegex Standard

Ces propriétés sont utilisées pour configurer le tFileInputRegex s’exécutant dans le framework de Jobs Standard.

Le composant tFileInputRegex Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Description</th>
</tr>
</thead>
</table>
| File name/Stream | **File name**: Chemin d’accès et nom du fichier, et/ou variable à traiter.  
| Row separator | Saisissez le séparateur utilisé pour identifier la fin d’une ligne. |
| Regex | Ce champ est compatible peut contenir plusieurs lignes. Intégrez à vos expressions régulières le subpattern correspondant aux champs à extraire.  
**Remarque** :  
En Java, doublez les antislashes en regexp. |
<p>| Header | Saisissez le nombre de lignes à ignorer au début du fichier. |
| Footer | Nombre de lignes à ignorer en fin de fichier. |
| Limit | Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n’est lue ou traitée. |</p>
<table>
<thead>
<tr>
<th><strong>Ignore error message for the unmatched record</strong></th>
<th>Cochez cette case afin de ne pas écrire en sortie les messages d’erreur pour les enregistrements qui ne correspondent pas à l’expression régulière définie. Par défaut, cette case n’est pas cochée.</th>
</tr>
</thead>
</table>
| **Schema et Edit Schema** | Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
  - **View schema** : sélectionnez cette option afin de voir le schéma.  
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquz sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| **Built-In** | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| **Repository** | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |
| **Skip empty rows** | Cochez cette case pour ignorer les lignes vides. |
| **Die on error** | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject. |
| **Advanced settings** |  |
| **Encoding** | Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.  
  Pour afficher cette liste dans la version Map/Reduce du tFileInputRegex, vous devez cocher la case Custom encoding. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant. |
Global Variables

| NB_LINE | nombre de lignes traitées. Cette variable est une variable After et retourne un entier. |
| ERROR_MESSAGE | message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

Utilisation

| Règle d’utilisation | Ce composant sert à lire un fichier et à en séparer les champs selon les Regex définies. Ce composant permet de créer un flux de données à l’aide d’un lien Row > Main, ainsi que de créer un flux de rejet avec un lien Row > Reject filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d’utilisation de ces deux liens, consultez Procédure à la page 1152 du composant tFileInputXML. |

Scénario : Transformer en fichier Regex en Positional

Le scénario suivant est un Job formé de deux composants, qui a pour objectif de lire les données d’un fichier d’entrée utilisant des expressions régulières et transformant les données délimitées en sortie positionnelle.

Déposer et relier les composants

Procédure

1. Cliquez et déposez un composant tFileInputRegex et tFileOutputPositional de la Palette dans l’espace de modélisation.
2. Cliquez-droit sur le composant tFileInputRegex et sélectionnez une connexion Row > Main dans le menu contextuel. Faites glisser cette connexion vers le composant tFileOutputPositional et relâchez lorsque le symbole approprié apparaît.
Configurer les composants

Procédure

1. Sélectionnez le tFileInputRegex à nouveau et définissez ses propriétés dans l’onglet Component :

2. Les propriétés de ce Job sont de type built-in. Par conséquent, elles ne sont définies que pour ce Job et ne peuvent être partagées avec d’autres utilisateurs.

3. Indiquez le chemin d’accès au fichier dans le champ File Name. Ce champ est obligatoire.

4. Définissez le séparateur de lignes, dans le champ Row separator, afin d’identifier la fin d’une ligne.

5. Puis, dans Regular expression, définissez l’expression régulière à utiliser pour délimiter les champs d’une ligne qui seront passés au composant suivant. Vous pouvez saisir une expression régulière multiligne si vous en avez besoin.

⚠️ Avertissement :

La syntaxe regex requiert des guillemets doubles.


7. Dans ce scénario, ignorez les champs en-tête (Header), pied de page (Footer) et limite (Limit).

8. Dans la liste Schema, sélectionnez local (Built-in) comme type de schéma des données à transférer au composant tFileOutputPositional.


10. Définissez les propriétés du second composant:
11. Saisissez le chemin d'accès au fichier positionnel de sortie.

12. Saisissez l'encodage du fichier de sortie dans le champ Encoding. Notez que pour l'instant, l'homogénéité de l'encodage dans un Job n'est pas vérifiée.


**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur Ctrl+S pour sauvegarder votre Job.

2. Passez à l'onglet Run et cliquez sur le bouton Run pour exécuter le Job.

Le fichier d'entrée est lu ligne par ligne puis divisé en champs délimités selon la valeur de l'expression régulière définie. Vous pouvez ouvrir le fichier de sortie avec tout éditeur standard.
tFileInputXML

Ce composant lit un fichier XML structuré ligne par ligne pour le scinder en champs et envoie les champs comme défini dans le schéma au composant suivant.

Propriétés du tFileInputXML Standard

Ces propriétés sont utilisées pour configurer le tFileInputXML s’exécutant dans le framework de Jobs Standard.

Le composant tFileInputXML Standard appartient aux familles File et XML.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
</tbody>
</table>

Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

| File name/Stream | File name : Chemin d’accès et nom du fichier à traiter.
Stream : Flux de données à traiter. Les données doivent préalablement être mises en flux afin d’être récupérées par le **tFileInputXML** via la variable représentative correspondante.
Cette variable peut être prédéfinie dans votre **Studio Talend** ou fournie par le contexte ou les composants utilisé(s) avec ce composant, par exemple la variable **INPUT_STREAM** du **tFileFetch**. Sinon, vous pouvez la définir manuellement et l’utiliser selon votre Job, par exemple à l’aide d’un **Java** ou d’un **JavaFlex**.
Afin d’éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d’autocomplétion (**Ctrl+Espace**) afin de remplir le champ, si cette variable a été correctement définie.
Pour plus d’informations concernant les flux d’entrée, consultez Scénario 2 : Lire les données d’un fichier distant en mode stream à la page 1072. |
| Loop XPath query | Nœud de l’arborescence sur lequel la boucle est basée. |
| Mapping | **Column** : Colonnes à mapper. Elles reflètent le schéma défini par le champ **Schema type**.
**XPath Query** : Saisissez les champs à extraire de l’entrée structurée.
**Get nodes** : Cochez cette case pour récupérer le contenu XML de tous les nœuds courants spécifiés dans le champ **XPath Query**, ou cochez la case correspondant à un nœud spécifique pour ne récupérer que le contenu du nœud sélectionné. Ces nœuds sont importants lorsque le flux de sortie de ce composant doit utiliser la structure XML, par exemple pour le type de données **Document**.
Pour plus d’informations, consultez le **Guide utilisateur du Studio Talend**.
**Remarque** :
L’option **Get Nodes** fonctionne en mode **DOM4j** et **SAX**, cependant avec le mode **SAX**, les espaces de nommage ne sont pas supportés. Pour plus d’informations concernant les modes **DOM4j** et **SAX**, consultez les propriétés de la liste **Generation mode** de l’onglet **Advanced settings**. |
| Limit | Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n’est lue ou traitée. Si -1, toutes les lignes sont lues ou traitées, quelque soit le nombre. |
| Die on error | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.
Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer... |
### Advanced settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ignore DTD file</strong></td>
<td>Cochez cette case afin d’ignorer le fichier DTD indiqué dans le fichier XML traité.</td>
</tr>
</tbody>
</table>
| **Advanced separator (for number)** | Cochez cette option pour modifier les séparateurs utilisés pour les nombres :  
| Thousands separator             | définitsez le séparateur utilisé pour les milliers.                        |
| Decimal separator               | définitsez le séparateur utilisé pour les décimaux.                         |
| **Ignore the namespaces**       | Cochez cette case pour ignorer les espaces de noms.                         |
| **Generate a temporary file**    | Renseignez le chemin d’accès et le nom du fichier temporaire XML.           |
| **Use Separator for mode Xerces** | Cochez cette case pour séparer les valeurs des nœuds fils concaténées.      |
| **Encoding**                    | Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données. |
| **Generation mode**             | Sélectionnez le mode de génération correspondant à votre mémoire disponible. Les modes disponibles sont :  
| Slow and memory-consuming - Dom4J | Cette option vous permet d’utiliser Dom4J pour traiter des fichiers XML très complexes.  
| Memory-consuming (Xerces)       | Consommateur de mémoire (Memory-consuming (Xerces))                        |
| Fast with low memory consumption (SAX) | Rapide et peu consommateur de mémoire (Fast with low memory consumption (SAX)) |
| **Validate date**               | Cochez cette case pour vérifier strictement le format de la date par rapport au schéma d’entrée. |
| **tStatCatcher Statistics**     | Cochez cette case pour collecter les données de log du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |
Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

Utilisation

| Règle d’utilisation | Ce composant doit être utilisé en tant que composant d’entrée. Il permet de créer un flux de données XML à l’aide d’un lien Row > Main, ainsi que de créer un flux de rejet avec un lien Row > Reject filtrant les données dont le type ne correspond pas au type défini. Pour un exemple d’utilisation de ces deux liens, consultez Procédure à la page 1152. |

Scénario 1 : Extraire des adresses XML

Ce scénario est constitué de deux composants : un composant tFileInputXML qui extrait des données du fichier XML contenant des adresses et un composant tLogRow qui affiche le résultat de cette opération dans la console Run via le composant tLogRow.

Procédure

1. Sélectionnez un composant tFileInputXML dans le dossier File de la Palette. Cliquez et déposez également un composant tLogRow de la famille Logs & Errors.
2. Connectez les deux composants à l’aide d’un lien Row > Main.
4. Le fichier XML contenant les adresses a été préalablement défini dans la zone **Metadata**, par conséquent récupérez ses propriétés en sélectionnant **Repository** dans **Property type**. De cette manière, les propriétés sont automatiquement récupérées et le reste des champs est renseigné (à l’exception du Schéma). Pour plus d’informations concernant l’assistant de création de métadonnées, consultez le **Guide utilisateur du Studio Talend**.

5. De la même manière, sélectionnez le schéma approprié dans la liste de métadonnées du **Repository**. Cliquez sur **Edit schema** si vous souhaitez modifier le schéma chargé.

6. Dans le champ **Filename**, indiquez le fichier structuré d’entrée.

7. Dans le champ **Loop XPath query**, modifiez le nœud de boucle si nécessaire.

8. Dans la table de **Mapping**, renseignez les champs à extraire et à afficher dans la sortie.

9. Si le fichier est volumineux, définissez un nombre de ligne maximum à traiter dans le champ **Limit**.

10. Saisissez l’encodage si nécessaire, puis double-cliquez sur **tLogRow** pour définir le caractère séparateur.

11. Enfin, appuyez sur **F6** ou cliquez sur le bouton **Run** dans la vue **Run** pour exécuter le Job. Dans la console, les champs définis sont extraits du fichier XML structuré et sont affichés.

**Résultats**

*Starting job XMLStreetFinder at 12:42 05/01/2007.*

Paris 2e arrondissement Rue de la Paix
Paris 8e arrondissement Champs Elysées
New York City Manhattan Madison avenue
New York City Brooklyn Washington heights

*Job XMLStreetFinder ended at 12:45 05/01/2007. jobit cokk*

**Scénario 2 : Extraire les données XML erronées dans un flux de rejet**

Ce scénario décrit un Job à trois composants qui permet de lire un fichier XML et :

1. d’une part, de retourner les données XML correctes dans un autre fichier XML,

2. et d’autre part, d’afficher dans la console de log les données erronées dont le type ne correspond pas au type défini.
Procédure

1. A partir de la Palette, glissez les composants tFileInputXML, tFileOutputXML et tLogRow dans l’espace de modélisation.
   Cliquez-droit sur le composant tFileInputXML, sélectionnez Row > Main dans le menu contextuel, et cliquez sur le composant tFileOutputXML pour créer le lien.
   Cliquez-droit sur le composant tFileInputXML, sélectionnez Row > Reject dans le menu contextuel, et cliquez sur le composant tLogRow pour créer le lien de rejet.

2. Double-cliquez sur le composant tFileInputXML pour afficher l’onglet Basic settings de la vue Component et paramétrer ses propriétés.

3. Dans la liste Property Type, sélectionnez Repository et cliquez sur le bouton [...] pour afficher la boîte de dialogue [Repository Content] à partir de laquelle vous pouvez sélectionner la métadonnée relative à votre fichier si vous l’avez stocké sous le nœud File xml du répertoire Metadata du Repository. Les champs suivants seront renseignés automatiquement. Sinon, sélectionnez le mode Built-in et renseignez ces champs manuellement.
Pour plus d’informations sur le stockage des schémas de métadonnées dans le Repository, consultez le Guide utilisateur du Studio Talend.

4. Dans le champ Schema, sélectionnez Repository et cliquez sur le bouton [...] pour sélectionner le schéma décrivant la structure de votre fichier si vous l’avez stocké dans le Repository. Sinon, cliquez sur le bouton [...] à côté du champ Edit schema et définissez le schéma manuellement.

Dans cet exemple, le schéma comprend cinq colonnes : id, CustomerName, CustomerAddress, idState et id2.

5. Cliquez sur le bouton [...] à côté du champ Filename et indiquez le chemin d’accès au fichier XML.

6. Dans le champ Loop XPath query, saisissez, entre guillemets, le chemin du nœud sur lequel effectuer une boucle.
   
   Dans la table Mapping, la colonne Column est automatiquement renseignée avec les colonnes définies dans le schéma.
   
   Dans la colonne XPath query, saisissez entre guillemets le nœud du fichier XML contenant les données que vous voulez extraire dans la colonne correspondante.

7. Dans le champ Limit, saisissez le nombre de lignes à traiter. Dans ce scénario, l’objectif est de traiter les 10 premières lignes.

8. Double-cliquez sur le composant tFileOutputXML pour afficher l’onglet Basic settings de la vue Component et paramétrer ses propriétés.

9. Cliquez sur le bouton [...] à côté du champ File Name pour sélectionner le répertoire de destination du fichier et saisissez le nom du fichier. Dans ce scénario, appelez le fichier customer_data.xml.
   
   Dans le champ Row tag, saisissez entre guillemets le nom que vous souhaitez donner à la balise contenant les données.

Dans cet exemple, le schéma comprend cinq colonnes : id, CustomerName, CustomerAddress, idState et id2.
Cliquez sur Edit schema et assurez-vous que le schéma est récupéré du schéma d’entrée. Si nécessaire, cliquez sur Sync columns pour récupérer le schéma du composant précédent.


Dans la zone Mode, sélectionnez l’option Vertical.


Résultats

Starting job tFileInputXML at 14:34 06/11/2009.

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>null</td>
</tr>
<tr>
<td>CustomerName</td>
<td>null</td>
</tr>
<tr>
<td>CustomerAddress</td>
<td>null</td>
</tr>
<tr>
<td>idState</td>
<td>null</td>
</tr>
<tr>
<td>id2</td>
<td>null</td>
</tr>
<tr>
<td>errorCode</td>
<td>null</td>
</tr>
<tr>
<td>errorMessage</td>
<td>Fcr input string: &quot;ab&quot; - line: 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>9</td>
</tr>
<tr>
<td>CustomerName</td>
<td>Childress Child Day Care</td>
</tr>
<tr>
<td>CustomerAddress</td>
<td>700 Tennyson Ave.</td>
</tr>
<tr>
<td>idState</td>
<td>12</td>
</tr>
<tr>
<td>id2</td>
<td>null</td>
</tr>
<tr>
<td>errorCode</td>
<td>null</td>
</tr>
<tr>
<td>errorMessage</td>
<td>Fcr input string: &quot;cd&quot; - line: 8</td>
</tr>
</tbody>
</table>

Le fichier customer_data.xml contenant les données correctes est créé et les données erronées sont affichées dans la console de la vue Run.
tFileList

Ce composant fait une boucle sur un jeu de fichiers ou de dossiers d’un répertoire donné en se basant sur un modèle de masque de fichiers.

Propriétés du tFileList Standard

Ces propriétés sont utilisées pour configurer le tFileList s’exécutant dans le framework de Jobs Standard.
Le composant tFileList Standard appartient aux familles File et Orchestration.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory</td>
<td>Chemin d’accès au répertoire de fichiers.</td>
</tr>
<tr>
<td>FileList Type</td>
<td>Sélectionnez à partir de la liste déroulante le type de source d’entrée utilisé : Files pour des fichiers, Directories pour des répertoires ou Both pour les deux types.</td>
</tr>
<tr>
<td>Include subdirectories</td>
<td>Cochez cette case si le ou les répertoires sélectionné(s) contiennent des sous-dossiers.</td>
</tr>
<tr>
<td>Case Sensitive</td>
<td>Cochez cette case pour créer un filtre selon la casse des noms des fichiers.</td>
</tr>
<tr>
<td>Generate Error if no file found</td>
<td>Cochez cette case pour être averti par un message d’erreur que la recherche de fichier n’a pas abouti.</td>
</tr>
<tr>
<td>Use Glob Expressions as Filemask (Unchecked means Perl5 Regex Expressions)</td>
<td>Cette case est cochée par défaut et permet de filtrer les résultats à l’aide d’expressions globales (Glob Expressions). Décochez la case pour filtrer les résultats à l’aide d’expressions de type regex Perl5.</td>
</tr>
<tr>
<td>Files</td>
<td>Cliquez sur le bouton [+] pour ajouter des lignes de filtre : Filemask : entrez un nom de fichier ou masque de fichier utilisant un caractère spécial ou une expression régulière.</td>
</tr>
<tr>
<td>Order by</td>
<td>Les dossiers sont listés en premier, puis viennent ensuite les fichiers. Vous pouvez définir les priorités des dossiers et des fichiers de différentes manières : By default : par ordre alphabétique, par dossier puis par fichier, By file name : par ordre alphabétique ou ordre alphabétique inverse, By file size : du plus petit léger au plus lourd ou du plus lourd au plus léger,</td>
</tr>
</tbody>
</table>
**Order action**

Choisissez de classer par ordre :

- **ASC** : ordre alphabétique / du plus petit au plus grand / du plus ancien au plus récent.
- **DESC** : à l'inverse de l'ordre alphabétique / du plus grand au plus petit / du plus ancien au plus récent.

**Advanced settings**

**Use Exclude Filemask**

Cochez cette case pour activer le champ Exclude Filemask afin d'exclure les conditions de filtre basées sur le type de fichier : Exclude Filemask. Renseignez le champ en saisissant les types de fichier devant être exclus du tableau Files de la vue Basic settings.

**Remarque** :

Les types de fichier dans ce champ doivent être entourés de guillemets doubles et séparés par des virgules.

**Format file path to slash (/) style (useful on Windows)**

Cochez cette case afin de formater le chemin d'accès au fichier en insérant des barres obliques (/), option utile sous Windows.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.

**Global Variables**

- **CURRENT_FILE** : nom du fichier courant. Cette variable est une variable Flow et retourne une chaîne de caractères.
- **CURRENT_FILEPATH** : nom du fichier courant ainsi que son chemin d'accès. Cette variable est une variable Flow et retourne une chaîne de caractères.
- **CURRENT_FILEEXTENSION** : extension du fichier courant. Cette variable est une variable Flow et retourne une chaîne de caractères.
- **CURRENT_FILEDIRECTORY** : répertoire du fichier courant. Cette variable est une variable Flow et retourne une chaîne de caractères.
**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Le composant <strong>tFileList</strong> fournit une liste de fichiers d’un répertoire donné sur lequel il fait une boucle.</th>
</tr>
</thead>
</table>
| Connections         | Liens de sortie (de ce composant à un autre) :  
                        **Row** : Iterate.  
                        Les liens d’entrée (d’un autre composant à celui-ci) :  
                        **Row** : Iterate.  
                        **Trigger** : Run If, OnSubjobOk, OnSubjobError, Synchronize, Parallelize.  
                        Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend. |

**Scénario 1 : Itération sur un répertoire**

Le scénario suivant décrit un Job de trois composants, qui a pour objectif de répertorier les fichiers d’un répertoire donné, de lire chaque fichier par itération, de sélectionner les données délimitées et d’afficher ces données en sortie dans la console **Run**.
Déposer et relier les composants

Procédure
1. Cliquez et déposez les composants suivants : tFileList, tFileInputDelimited et tLogRow dans l'espace de modélisation.
2. Cliquez-droit sur le composant tFileList et connectez-le au composant tFileInputDelimited via une connexion de type Row Iterate. Puis connectez le tFileInputDelimited au tLogRow à l'aide d'une connexion de type Row Main.

Configurer les composants

Procédure
1. Sélectionnez le composant tFileList et cliquez sur l'onglet Basic settings :

   "D:/Java/Files/Input/tfileList"(tFileList_1)

   Directory
   FileList Type Files
   Filemask *
   Case sensitive Yes
   Use Glob Expressions as Filemask

2. Dans le champ Directory, indiquez le chemin d'accès des fichiers à traiter. Pour faire apparaître ce chemin d'accès dans le Job lui-même, utilisez le nom de champ (__DIRECTORY__) qui apparaît en bulle info lorsque vous passez le souris sur le champ Directory. Saisissez cette référence dans le champ Label Format de l'onglet View.

3. Dans le champ FileList Type, sélectionnez l'option Files.
4. Définissez le champ Case sensitive sur Yes pour prendre la casse en compte dans cet exemple.
5. Laissez la case Use Glob Expressions as Filemask cochée si vous utilisez des expressions globales pour filtrer les fichiers et définissez un masque de fichier dans le champ Filemask.
6. Double-cliquez sur le tFileInputDelimited afin d'ouvrir sa vue Basic settings et définissez ses propriétés.
7. Dans le champ **File Name**, nommez le fichier à l’aide d’une variable contenant son chemin d’accès, tel que vous l’avez renseigné dans les propriétés du **tFileList**. Appuyez sur **Ctrl+Espace** pour accéder à la liste d’autocomplétion des variables et sélectionnez la variable globale `((String)globalMap.get("tFileList_1_CURRENT_FILEPATH"))`.

8. Renseignez les autres champs comme pour un fichier délimité **tFileInputDelimited** standard. Voir également : **tFileInputDelimited** à la page 1067.

9. Sélectionnez le composant de sortie **tLogRow** et renseignez le séparateur de champs qui s’affichera dans la console de l’onglet **Run**. Voir également : **Propriétés du tLogRow Standard** à la page 2105.

### Sauvegarder et exécuter le Job

Appuyez sur **Ctrl+S** pour sauvegarder votre Job et appuyez sur **F6** pour l’exécuter.

```
Job test ended at 17:16 21/09/2009. (exit code=0)
```
L’itération s’effectue sur le répertoire défini. Chacun des fichiers contenus dans ce répertoire est lu. Puis les données délimitées sont transmises au composant de sortie qui les affiche dans la console Log de la vue Run.

**Scénario 2 : Trouver des fichiers dupliqués entre deux dossiers**

Le scénario suivant décrit un Job qui fait une boucle sur deux dossiers, transforme les résultats de l’itération en flux de données afin d’obtenir une liste de noms de fichiers puis affiche les doublons dans la console Run. Ce scénario peut servir de préparation pour la fusion de deux dossiers par exemple.

![Diagramme scénario 2](image)

**Déposer et relier les composants**

**Procédure**

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : deux tFileList, deux tIterateToFlow, deux tFileOutputDelimited, un tFileInputDelimited, un tUniqRow et un tLogRow.

2. Reliez le premier tFileList au premier tIterateToFlow à l’aide d’un lien Row > Iterate et reliez le premier tIterateToFlow au premier tFileOutputDelimited à l’aide d’un lien Row > Main afin de former le premier sous-Job.

3. Reliez le second tFileList au second tIterateToFlow à l’aide d’un lien Row > Iterate et reliez le second tIterateToFlow au second tFileOutputDelimited à l’aide d’un lien Row > Main afin de former le deuxième sous-Job.

4. Reliez le tFileInputDelimited au tUniqRow à l’aide d’un lien Row > Main et reliez le tUniqRow au tLogRow à l’aide d’un lien Row > Duplicates afin de former le troisième sous-Job.

Configurer les composants

Procédure

1. Dans le champ Directory de la vue Basic settings du premier tFileList, saisissez le chemin vers le premier dossier dans lequel vous souhaitez lire le nom des fichiers, E:/DataFiles/DI/images dans ce scénario. Laissez les autres paramètres tels qu’ils sont.

2. Double-cliquez sur le premier tIterateToFlow afin d’afficher sa vue Basic settings.

3. Double-cliquez sur le bouton [...] situé à côté de Edit schema afin d’ouvrir la boîte de dialogue [Schema]. Définissez le schéma du fichier texte dans lequel le second composant doit écrire les noms des fichiers. Une fois fait, cliquez sur OK afin de fermer la boîte de dialogue et propager les changements au composant suivant.

Dans cet exemple, le schéma contient une seule colonne : Filename.
4. Dans le champ **Value** de la table Mapping, appuyez sur **Ctrl+Espace** afin d’ouvrir la liste d’autocomplétion des variables. Sélectionnez la variable globale `((String)globalMap.get("tFileList_1_CURRENT_FILE"))` afin de lire le nom de chaque fichier du dossier d’entrée. Ce dernier sera ensuite inséré dans un flux de données et passé au composant suivant.

5. Dans le champ **File Name** de la vue **Basic settings** du premier **tFileOutputDelimited**, saisissez le chemin vers le fichier texte contenant les noms des fichiers provenant du flux entrant, **D:/temp/tempdata.csv** dans cet exemple. La configuration du premier sous-Job est ainsi terminée.

6. Configurez le deuxième sous-Job de la même manière que le premier sous-Job. Cependant :
   - Dans le champ **Directory** de la vue **Basic settings** du second **tFileList**, saisissez le chemin vers l’autre dossier dans lequel vous souhaitez lire les noms des fichiers, **E:/DataFiles/DQ/images** dans ce scénario.
   - Dans la vue **Basic settings** du second **tFileOutputDelimited**, cochez la case **Append** afin que les noms des fichiers précédemment écrits dans le fichier texte ne soit pas écrasés.

7. Dans le champ **File name/Stream** de la vue **Basic settings** du **tFileInputDelimited**, saisissez le chemin vers le fichier texte contenant la liste de noms de fichiers, **D:/temp/tempdata.csv** dans ce scénario. Définissez le schéma du fichier qui, dans le présent exemple, contient une seule colonne, **Filename**.
8. Dans la vue **Basic settings** du **tUniqRow**, cochez la case **Key attribute** pour la colonne, *Filename* in this example.

9. Dans la vue **Basic settings** du **tLogRow**, sélectionnez l’option **Table (print values in cells of a table)** pour un meilleur affichage des résultats d’exécution du Job.

**Exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** afin de sauvegarder le Job.
2. Cliquez sur **Run** ou appuyez sur **F6** afin d’exécuter le Job.
   
   Tous les fichiers dupliqués entre les deux dossiers sélectionnés sont affichés dans la console.
Résultats

Pour d'autres scénarios utilisant le **tFileList**, **tFileCopy** à la page 1039.
tFileOutputARFF

Ce composant écrit un fichier ARFF contenant des données organisées en fonction du schéma défini.

Propriétés du tFileOutputARFF Standard

Ces propriétés sont utilisées pour configurer le tFileOutputARFF s’exécutant dans le framework de Jobs Standard.
Le composant tFileOutputARFF Standard appartient à la famille File.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>File name</td>
<td>Chemin d’accès et nom du fichier de sortie, et/ou variable à utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Attribute Define</td>
<td>Reflète le schéma tel qu’il est défini dans la boîte de dialogue [Edit schema].</td>
</tr>
<tr>
<td></td>
<td>Column : Nom de la colonne.</td>
</tr>
<tr>
<td></td>
<td>Type : Type des données de la colonne.</td>
</tr>
<tr>
<td></td>
<td>Pattern : Saisissez le modèle (pattern) de la donnée si nécessaire.</td>
</tr>
<tr>
<td>Relation</td>
<td>Saisissez le nom de la relation.</td>
</tr>
<tr>
<td>Append</td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository.</td>
</tr>
</tbody>
</table>
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


**Create directory if not exists**

Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s’il n’existe pas déjà.

**Advanced settings**

**Don’t generate empty file**

Cochez cette case pour empêcher la génération d’un fichier vide.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Global Variables**

**Global Variables**

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace**.
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Utilisez ce composant pour écrire dans un fichier ARFF des données récupérées d’autres composants à l’aide d’une connexion de type Row.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
| Connections        | Liens de sortie (de ce composant à un autre) :
|                    | **Row** : Main. |
|                    | **Trigger** : On Subjob Ok; On Subjob Error; Run if. |
|                    | Liens d’entrée (d’un autre composant à celui-ci) :
|                    | **Row** : Main; Reject; Iterate. |
|                    | **Trigger** : On Subjob Ok; On Subjob Error; Run if; On Component Ok; On Component Error; Synchronize; Parallelize. |
| Limitation         | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour |
ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

**Scénario associé**

Pour un scénario associé, consultez **Scénario : Afficher le contenu d’un fichier ARFF** à la page 1064.
tFileOutputDelimited

Ce composant écrit en sortie les données d’entrée dans un fichier délimité en respectant le schéma défini.

Propriétés du tFileOutputDelimited Standard

Ces propriétés sont utilisées pour configurer le tFileOutputDelimited s’exécutant dans le framework de Jobs Standard.

Le composant tFileOutputDelimited Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Property type                  | Peut être **Built-In** ou **Repository**.
|-------------------------------|---------------------------------------------------------------
| **Built-In**                  | propriétés utilisées ponctuellement.
| **Repository**                | sélectionnez le fichier dans lequel sont stockées les propriétés du composant.

**Use Output Stream**

Cochez la case pour traiter le flux de données qui vous intéresse. Une fois cochée, le champ **Output Stream** s’affiche et vous pouvez saisir le flux de données souhaité.

Le flux de données à traiter doit être ajouté au flux afin que ce composant récupère ces données via la variable représentative correspondante.

Cette variable peut être prédéfinie dans votre Studio ou fournie par le contexte ou les composants utilisé(s) avec ce composant. Sinon, vous pouvez la définir manuellement et l’utiliser selon votre Job, par exemple à l’aide d’un **tJava** ou d’un **tJavaFlex**.

Afin d’éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d’autocomplétion (Ctrl+Espace) afin de remplir le champ, si cette variable a été correctement définie.

Pour plus d’informations concernant l’utilisation d’un flux, consultez Scénario 2 : Lire les données d’un fichier distant en mode stream à la page 1072.

**File name**

Chemin d’accès et nom du fichier de sortie et/ou variable à utiliser.

Ce champ est indisponible lorsque vous avez cochée la case **Use Output Stream**.

Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend.
| **Row Separator** | Saisissez le séparateur utilisé pour identifier la fin d'une ligne. |
| **Field Separator** | Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées. |
| **Append** | Cochez cette option pour ajouter de nouvelles lignes à la fin du fichier. |
| **Include Header** | Cochez cette case pour tenir compte de l'en-tête dans le fichier de sortie. |
| **Compress as zip file** | Cochez cette case pour compresser en zip le fichier de sortie. |

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

- **Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

### Sync columns

Cliquez sur ce bouton pour synchroniser le schéma de sortie avec celui d’entrée. La fonction **Sync** ne s’affiche que si une connexion de type **Row** est liée au composant de sortie.

### Advanced settings

| **Advanced separator (for numbers)** | Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers |
Il est une virgule (,) et le séparateur décimal est un point (.).

**Thousands separator** : définissez le séparateur utilisé pour les milliers.

**Decimal separator** : définissez le séparateur utilisé pour les décimaux.

| CSV options | Cochez cette case afin de spécifier les paramètres CSV suivants :

  - **Escape char** : saisissez le caractère d'échappement entre guillemets doubles. Par exemple, vous devez saisir "\" si le caractère barre oblique inversée (/) est utilisé comme caractère d'échappement.

  - **Text enclosure** : saisissez le caractère avec lequel entourer le texte (un seul caractère) entre guillemets doubles. Par exemple, vous devez saisir "" lorsque le guillemet double (") est utilisé comme caractère pour entourer le texte.

| Create directory if not exists | Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s’il n’existe pas déjà.

| Split output in several files | Si le fichier délimité en sortie est volumineux, vous pouvez le scinder en plusieurs fichiers.

  - **Rows in each output file** : saisissez le nombre de lignes pour chaque fichier de sortie.

| Custom the flush buffer size | Cochez cette case pour personnaliser le nombre de lignes à écrire avant de vider le tampon.

  - **Row Number** : Nombre de lignes à écrire avant de vider le tampon.

| Output in row mode | Cochez cette case pour assurer que l’action de vider le tampon se termine complètement (atomicité), afin que chaque ligne de données reste cohérente en tant qu’ensemble et que les lignes incomplètes de données ne soient jamais écrites dans un fichier.

  Cette case est principalement utile lorsque vous utilisez ce composant en multithread.

| Encoding | Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.

| Don’t generate empty file | Cochez cette case pour empêcher la génération d’un fichier vide.

| Throw an error if the file already exist | Cochez cette case pour retourner une exception si le fichier de sortie spécifié dans le champ **File Name**, dans l’onglet **Basic settings** existe déjà.

  Décochez cette case pour écraser le fichier existant.
## tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FILE_NAME : nom du fichier traité. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Utilisez ce composant pour écrire dans un fichier délimité des données récupérées d’autres composants à l’aide d’une connexion de type Row et séparer les champs à l’aide d’un séparateur.</th>
</tr>
</thead>
</table>

### Scénario 1 : Ecrire des données dans un fichier délimité

Ce scénario est constitué de trois composants et a pour objectif d’extraire certaines données d’un fichier contenant des informations sur les clients (customers) puis de les écrire dans un fichier délimité en sortie.
Dans l'exemple suivant, vous avez enregistré les informations de connexion et les métadonnées dans le Repository afin de faciliter leur réutilisation. Pour plus d'informations sur comment centraliser les données de connexion et autres métadonnées d'un fichier dans le Repository, consultez le Guide utilisateur du Studio Talend.

Déposer et relier les composants

Procédure

1. Dans le Repository, ouvrez le nœud File Delimited de la catégorie Metadata. Cliquez sur la métadonnée customers, et déposez-la dans l'espace de modélisation graphique. Une boîte de dialogue s'ouvre et vous demande quel type de composant vous souhaitez déposer.

2. Cliquez sur tFileInputDelimited puis sur OK. Un composant tFileInputDelimited contenant le nom de votre schéma d'entrée s'affiche dans l'espace de modélisation graphique.

3. Déposez un composant tMap et un composant tFileOutputDelimited de la Palette dans l'espace de modélisation graphique.

4. Reliez les composants à l'aide d'un lien Row > Main.

Configurer les composants

Configurer le composant d'entrée

Procédure

1. Double-cliquez sur le composant tFileInputDelimited afin d'afficher sa vue Basic settings. Tous les champs sont renseignés automatiquement car vous avez défini votre fichier d'entrée localement.
2. Si vous ne définissez pas votre fichier d’entrée localement dans la vue **Repository**, renseignez les détails manuellement après avoir sélectionné **Built-in** dans la liste **Property Type**.

3. Cliquez sur le bouton [..] près du champ **File Name** et dans la boîte de dialogue de sélection, parcourrez le système de fichiers et sélectionnez le fichier d’entrée, dans cet exemple: *customer.csv*.

   **Avertissement :**

   Si le chemin d’accès du fichier contient des caractères accentués, vous obtiendrez un message d’erreur lors de l’exécution du Job.

4. Dans les champs **Row Separator** et **Field Separator**, saisissez respectivement le séparateur de ligne s “\n” et le séparateur de champs “;”.

5. Si besoin, définissez dans les champs correspondants le nombre de lignes utilisées dans l’en-tête (**Header**) et dans le pied-de-page (**Footer**), puis définissez le nombre de lignes traitées. Dans cet exemple, saisissez 6 pour **Header** et laissez par défaut les champs **Footer** et **Limit**.

6. Dans le champ **Schema**, le schéma est défini automatiquement sur **Repository** et votre schéma est déjà défini puisque vous avez stocké votre fichier d’entrée localement pour cet exemple. Sinon, sélectionnez **Built-in** et cliquez sur le bouton [..] près de Edit Schema pour ouvrir la boîte de dialogue [Schema], dans laquelle vous pouvez définir le schéma d’entrée. Cliquez ensuite sur **OK** pour fermer la boîte de dialogue.
Configurer le composant de mapping

Procédure

1. Double-cliquez sur le tMap afin d’ouvrir son éditeur.

2. Dans cet éditeur, cliquez sur le bouton en haut à droite pour ouvrir la boîte de dialogue [Add a new output table].

3. Saisissez un nom pour la table que vous souhaitez créer, row2 dans cet exemple.
4. Cliquez sur OK pour valider vos modifications et fermer la boîte de dialogue.
5. Dans la table de gauche (row1), sélectionnez les trois premières lignes (Id, CustomerName et CustomerAddress) en appuyant sur Ctrl, puis glissez-les dans la table de droite.
6. Dans le Schema editor en bas de l’éditeur, changez le Type de la ligne RegisterTime, dans la table de droite, et passez-le en chaîne de caractères, String.

7. Cliquez sur OK pour enregistrer les modifications et fermer l’éditeur.

Configurer le composant de sortie

Procédure

1. Dans l’espace de modélisation, double-cliquez sur le composant tFileOutputDelimited pour ouvrir sa vue Component et définir ses propriétés.
2. Dans le champ **Property Type**, sélectionnez le mode **Built-in** et renseignez les champs suivants manuellement.

3. Cliquez sur le bouton [...] correspondant au champ **File Name**. Dans la boîte de dialogue, indiquez le répertoire de destination et le nom du fichier dans lequel vous voulez écrire des données, ici *customerselection.txt*.

4. Pour les champs **Row Separator** et **Field Separator**, saisissez respectivement le séparateur de ligne "\n" et le séparateur de champs ";". Ils sont utilisés dans le fichier de sortie pour délimiter les valeurs et les lignes.

5. Cochez la case **Include Header** si vous souhaitez récupérer le libellé des colonnes.

6. Cliquez sur **Edit schema** et assurez-vous que le schéma est correctement récupéré du schéma d’entrée. Si nécessaire, cliquez sur **Sync Columns** pour récupérer le schéma du composant précédent.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur **Run** dans l’onglet **Run** pour exécuter le Job.

```plaintext
1 id;CustomerName;CustomerAddress
2 1;Griffith Paving and Sealing;talend@apres91
3 2;Bill's Dive Shop;511 Maple Ave. Apt. 1B
4 3;Childress Child Day Care;662 Lyons Circle
5 4;Faceiif Kitchen and Bath;unknown
6 5;Terrini & Son Auto and Truck;770 Exmoor Rd.
7 6;Kermit the Pet Shop;1060 Parkside Ln.
8 7;Tub's Furniture Store;307 Old Trail Rd.
9 8;Toglette & Myerson Ltd;610 Sheridan Rd.
10 9;Childress Child Day Care;788 Tennyson Ave.
11 10;Elle Hypnosis and Therapy Cent;2032 Northbrook Ct.
```

Le fichier créé est délimité par des points-virgules, il ne contient que les colonnes *Id, CustomerName* et *CustomerAddress*.

Les colonnes *Id, CustomerName* et *CustomerAddress* sont écrites dans le fichier de sortie défini.
Scénario 2 : Utiliser un flux de sortie pour sauvegarder des données dans un fichier local

Basé sur le scénario précédent, ce scénario sauvegarde les données filtrées dans un fichier local, à l’aide de l’option Output Stream.

Déposer et relier les composants

Procédure
1. Déposez un composant tJava de la Palette dans l’espace de modélisation graphique.
2. Connectez le tJava au tFileInputDelimited à l’aide d’un lien Trigger > On Subjob Ok.

Configurer les composants

Procédure
1. Double-cliquez sur le tJava pour ouvrir sa vue Basic settings et configurer ses propriétés.
2. Dans la zone Code, saisissez la commande suivante :

   ```java
   new java.io.File("C:/myFolder").mkdirs();
globalMap.put("out_file",new java.io.FileOutputStream("C:/myFolder/customerselection.txt",false));
   ```

   Remarque :
   Dans ce scénario, la commande utilisée dans la zone Code du tJava va créer un nouveau dossier C:/myFolder dans lequel le fichier de sortie customerselection.txt sera sauvegardé. Vous pouvez personnaliser la commande selon vos besoins.
3. Double-cliquez sur le tFileOutputDelimited afin d’ouvrir sa vue Basic settings et configurer ses propriétés.
4. Cochez la case **Use Output Stream** pour activer le champ **Output Stream**, dans lequel vous pouvez définir le flux de sortie, à l’aide d’une commande.

Renseignez le champ **Output Stream** en saisissant la commande suivante :

```
(java.io.OutputStream)globalMap.get("out_file")
```

**Remarque :**
Vous pouvez personnaliser la commande dans le champ **Output Stream** en appuyant sur les touches **Ctrl+Espace** pour utiliser l’auto-complétion ou saisissez votre commande manuellement dans le champ. Dans ce scénario, la commande utilisée dans le champ **Output Stream** appelle la classe `java.io.OutputStream` pour écrire le flux de données filtrées dans un fichier local, défini dans la zone **Code** du composant **tJava**.

5. Cliquez sur **Sync columns** afin de récupérer le schéma du composant précédent.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l’onglet **Run** pour l’exécuter.

Les trois colonnes spécifiées, *Id*, *CustomerName* et *CustomerAddress* sont écrites dans le fichier de sortie défini.
tFileOutputExcel

Ce composant écrit un fichier MS Excel avec des valeurs de données séparées, en fonction du schéma défini.

Propriétés du tFileOutputExcel Standard

Ces propriétés sont utilisées pour configurer le tFileOutputExcel s'exécutant dans le framework de Jobs Standard.

Le composant tFileOutputExcel Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property (tFileOutputExcel)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write excel 2007 file format (xlsx)</td>
<td>Cochez cette case pour écrire les données traitées au format Excel 2007, .xlsx.</td>
</tr>
</tbody>
</table>
| Use Output Stream | Cochez la case pour traiter le flux de données qui vous intéresse. Une fois cochée, le champ Output Stream s’affiche et vous pouvez saisir le flux de données souhaité. 
Le flux de données à traiter doit être ajouté au flux afin que ce composant récupère ces données via la variable représentative correspondante. 
Cette variable peut être prédéfinie dans votre Studio ou fournie par le contexte ou les composants utilisé(s) avec ce composant. Sinon, vous pouvez la définir manuellement et l’utiliser selon votre Job, par exemple à l’aide d’un tJava ou d’un tJavaFlex. 
Afin d’éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d’autocomplétion (Ctrl+Espace) afin de remplir le champ, si cette variable a été correctement définie. 
Pour plus d’informations concernant l’utilisation d’un flux, consultez Scénario 2 : Lire les données d’un fichier distant en mode stream à la page 1072. |
| File name | Chemin d’accès et nom du fichier de sortie. 
Ce champ est indisponible lorsque vous avez cochée la case Use Output Stream. 
Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend. |
<p>| Sheet name | Nom de la feuille. |
| Include header | Cochez cette case pour tenir compte de l’en-tête dans le fichier de sortie. |
| Append existing file | Cochez cette option pour ajouter de nouvelles lignes à la fin du fichier. |</p>
<table>
<thead>
<tr>
<th><strong>Append existing sheet</strong></th>
<th>Cochez cette option pour ajouter de nouvelles lignes à la fin de la feuille Excel.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Is absolute Y pos.</strong></td>
<td>Cochez cette case pour rajouter des informations dans des cellules préexistantes :</td>
</tr>
<tr>
<td><strong>First cell X</strong></td>
<td>emplacement de la cellule sur l’axe des abscisses.</td>
</tr>
<tr>
<td><strong>First cell Y</strong></td>
<td>emplacement de la cellule sur l’axe des ordonnées.</td>
</tr>
<tr>
<td><strong>Keep existing cell format</strong></td>
<td>lorsque la case est cochée, cette option permet de garder le format et l’alignement de la cellule.</td>
</tr>
<tr>
<td><strong>Font</strong></td>
<td>Sélectionnez dans la liste déroulante la police à utiliser.</td>
</tr>
<tr>
<td><strong>Define all columns auto size</strong></td>
<td>Cette case est décochée par défaut, vous pouvez alors préciser dans le tableau associé quelles sont les colonnes à définir automatiquement. Cochez cette case pour que la taille de toutes les colonnes soit définies automatiquement.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être <strong>Built-in</strong> ou distant dans le <strong>Repository</strong>. Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <strong>No</strong> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Sync columns</strong></td>
<td>Cliquez sur ce bouton pour synchroniser le schéma de sortie avec celui d’entrée. La fonction <strong>Sync</strong> ne s’affiche que si une connexion de type <strong>Row</strong> est liée au composant de sortie.</td>
</tr>
</tbody>
</table>
### Advanced settings

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Create directory if not exists</strong></td>
<td>Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s’il n’existe pas déjà.</td>
</tr>
<tr>
<td><strong>Custom the flush buffer size</strong></td>
<td>Disponible lorsque la case <strong>Write excel2007 file format</strong> est cochée, dans la vue <strong>Basic settings</strong>. Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ <strong>Row number</strong>, saisissez le nombre de lignes après lesquelles la mémoire est à nouveau libérée.</td>
</tr>
</tbody>
</table>
| **Advanced separator (for numbers)**       | Cochez cette option pour modifier les séparateurs utilisés pour les nombres :  

- **Thousands separator** : définissez le séparateur utilisé pour les milliers.
- **Decimal separator** : définissez le séparateur utilisé pour les décimaux.                                                                                                                                                                                                                                                                 |
| **Encoding**                                | Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.                                                                                                                      |
| **Don’t generate empty file**               | Cochez cette case pour ne pas générer de fichier vide.                                                                                                                                                                                                                                                                             |
| **Recalculate formula**                     | Cochez cette case si vous avez besoin de recalculer une ou des formule(s) dans le fichier Excel spécifié. Cette case apparaît uniquement lorsque vous cochez ces trois cases : Write excel2007 file format(xlsx), Append existing file et Append existing sheet.                                                                                          |
| **tStatCatcher Statistics**                 | Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.                                                                                                                                                                                                       |

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette
liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Utilisez ce composant pour écrire dans un fichier MS Excel des données récupérées d’autres composants à l’aide d’une connexion de type Row.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé, consultez tSugarCRMInput à la page 4014.

Pour un scénario relatif à l’utilisation de l’option Use Output Stream, consultez Scénario 2 : Utiliser un flux de sortie pour sauvegarder des données dans un fichier local à la page 1177.
tFileOutputJSON

Ce composant reçoit des données et les réécrit dans un bloc de données structurées JSON, dans un fichier de sortie.

Propriétés du tFileOutputJSON Standard

Ces propriétés sont utilisées pour configurer le tFileOutputJSON s’exécutant dans le framework de Jobs Standard.

Le composant tFileOutputJSON Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>File Name</th>
<th>Nom du fichier de sortie et son chemin d’accès.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate an array json</td>
<td>Cochez cette case pour générer un fichier de tableau JSON.</td>
</tr>
<tr>
<td>Name of data blocks</td>
<td>Saisissez, entre guillemets doubles, un nom pour le bloc de données à écrire. Ce champ disparaît lorsque la case Generate an array json est cochée.</td>
</tr>
</tbody>
</table>

Schema et Edit Schema

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.


**Sync columns**

Cliquez pour synchroniser le schéma du fichier de sortie et le schéma du fichier d'entrée. La fonction **Sync** ne s'affiche que lorsqu'un lien **Row** connecte le composant de sortie.

**Advanced settings**

**Create directory if not exists**

Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant.

**Global Variables**

**Global Variables**

- **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.

- **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. 
Utilisation

**Règle d'utilisation**

Utilisez ce composant pour réécrire des données reçues dans un fichier de sortie structuré JSON.

---

**Scénario : Ecrire un fichier JSON structuré**

Ce scénario de deux composants comprend un composant tRowGenerator qui génère des données au hasard, ainsi qu'un composant tFileOutputJSON qui écrit dans un fichier structuré JSON de sortie.

---

**Procédure**

1. Glissez de la Palette dans l'espace de modélisation graphique un tRowGenerator et un tFileOutputJSON.
2. Liez les composants à l'aide d'un lien **Row > Main**.
3. Double-cliquez sur le tRowGenerator pour définir ses **Basic settings** dans la vue **Component**.

   ![tRowGenerator_1](image)

   **Basic settings**

4. Cliquez sur [...] à côté du bouton **Edit Schema** afin d'afficher la boîte de dialogue et définir son schéma.

   ![Edit Schema](image)

5. Cliquez sur le bouton [+ ] pour ajouter les colonnes désirées.

---

1185
7. Sous **Type**, sélectionnez le type dans la liste.
8. Cliquez sur **OK** pour fermer la boîte de dialogue.
9. Cliquez sur le bouton [*+] à côté de **RowGenerator Editor** pour ouvrir la boîte de dialogue correspondante.

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Type</th>
<th>N.</th>
<th>Functions</th>
<th>Environment</th>
<th>variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>getLastName</td>
<td></td>
</tr>
<tr>
<td>age</td>
<td></td>
<td>Integer</td>
<td></td>
<td>random</td>
<td>min_value=10; max...</td>
<td></td>
</tr>
<tr>
<td>birthplace</td>
<td></td>
<td>String</td>
<td></td>
<td>...</td>
<td>&quot;London&quot;, &quot;Manchester...</td>
<td></td>
</tr>
</tbody>
</table>

10. Sous **Functions**, sélectionnez les fonctions pré-définies pour les colonnes, si besoin, ou sélectionnez [...] pour définir les paramètres personnalisés des fonctions, dans l'onglet **Function parameters**.
11. Saisissez le nombre de lignes à générer, dans le champ **Number of Rows for RowGenerator**.
12. Cliquez sur **OK** pour fermer la boîte de dialogue.
13. Cliquez sur le composant **tFileOutputJSON** afin de définir ses propriétés, dans l'onglet **Basic settings** de la vue **Component**.

14. Cliquez sur le bouton [...] pour parcourir votre répertoire et définir où vous souhaitez générer le fichier de sortie JSON, puis saisissez un nom de fichier.
15. Saisissez un nom pour le bloc de données à générer dans le champ correspondant, entre guillemets doubles.
16. Sélectionnez **Built-In** dans la liste **Schema**.
17. Cliquez sur **Sync Columns** afin de retrouver le schéma du précédent composant.
18. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
Appuyez sur F6 ou cliquez sur le bouton Run de l'onglet Run pour exécuter le Job.

```
{
  "person": [
    {
      "birthplace": "Manchester",
      "age": 49,
      "name": "Carter"
    },
    {
      "birthplace": "Liverpool",
      "age": 39,
      "name": "Clinton"
    },
    {
      "birthplace": "London",
      "age": 53,
      "name": "Taylor"
    }
  ]
}
```

**Résultats**

Les données du schéma d'entrée sont écrites dans un bloc de données structurées JSON, dans le fichier de sortie.
tFileOutputLDIF

Ce composant écrit ou modifie un fichier LDIF avec des données séparées en différentes entrées selon le schéma défini, ou supprime le contenu d’un fichier LDIF.

Le tFileOutputLDIF extrait des données vers un fichier LDIF qui peut ensuite être chargé dans un annuaire LDAP.

Propriétés du tFileOutputLDIF Standard

Ces propriétés sont utilisées pour configurer le tFileOutputLDIF s’exécutant dans le framework de Jobs Standard.

Le composant tFileOutputLDIF Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>File Name</th>
<th>Spécifiez le chemin d’accès au fichier LDIF de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrap</td>
<td>Spécifiez le nombre de caractères avant un retour à la ligne.</td>
</tr>
<tr>
<td>Change type</td>
<td>Sélectionnez un Changetype pour spécifier le type d'opération à effectuer sur les entrées du fichier LDIF de sortie.</td>
</tr>
<tr>
<td></td>
<td>• Add : opération LDAP ajoutant une entrée.</td>
</tr>
<tr>
<td></td>
<td>• Modify : opération LDAP modifiant une entrée.</td>
</tr>
<tr>
<td></td>
<td>• Delete : opération LDAP supprimant une entrée.</td>
</tr>
<tr>
<td></td>
<td>• Modrdn : opération LDAP modifiant le RDN (Relative Distinguished Name) d'une entrée.</td>
</tr>
<tr>
<td></td>
<td>• Default : opération LDAP par défaut.</td>
</tr>
<tr>
<td>Multi-Values / Modify Detail</td>
<td>Spécifiez les attributs pour les champs multi-valués lorsque l’option Add ou Default est sélectionnée dans la liste Change type ou fournissez les informations détaillées des modifications lorsque l’option Modify est sélectionnée dans la liste Change type.</td>
</tr>
<tr>
<td></td>
<td>• Column : les cellules de la colonne Column sont automatiquement renseignées avec le nom des colonnes définies dans le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Operation : sélectionnez une opération à effectuer sur le champ correspondant. Cette colonne est disponible uniquement lorsque l’option Modify est sélectionnée dans la liste Change type.</td>
</tr>
<tr>
<td></td>
<td>• MultiValue : cochez la case si le champ correspondant est un champ multi-valué.</td>
</tr>
<tr>
<td></td>
<td>• Separator : spécifiez le séparateur de valeurs dans le champ multi-valué correspondant.</td>
</tr>
</tbody>
</table>
• **Binary** : cochez la case si le champ correspondant représente des données binaires.

• **Base64** : cochez la case si le champ correspondant doit être encodé en base-64. Les données encodées en base-64 dans le fichier LDIF sont représentées par le symbole `:`.

Cette table est disponible lorsque l’option **Add, Modify**, ou **Default** est sélectionnée dans la liste **Change type**.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

• **View schema** : sélectionnez cette option afin de voir le schéma.

• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Built-In

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

### Repository

- **Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

### Sync columns

- **Sync columns** : Cliquez pour synchroniser le schéma du fichier de sortie avec le schéma du fichier d'entrée. La fonction **Sync** n’apparaît qu’une fois la connexion Row mise en place entre les composants d’entrée et de sortie.

### Append

- **Append** : Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.

### Advanced settings

<table>
<thead>
<tr>
<th>Enforce safe base 64 conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select this check box to enable the safe base-64 encoding. For more detailed information about the safe base-64 encoding, see <a href="https://www.ietf.org/rfc/rfc2849.txt">https://www.ietf.org/rfc/rfc2849.txt</a>.</td>
</tr>
<tr>
<td>Create directory if not exists</td>
</tr>
<tr>
<td>--------------------------------</td>
</tr>
<tr>
<td>Custom the flush buffer size</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Row number</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Encoding</td>
</tr>
<tr>
<td>Don’t generate empty file</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Utilisez ce composant pour écrire dans un fichier LDIF des données provenant d’un composant d’entrée utilisant un lien Row &gt; Main.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton Install dans l’onglet Component. Vous pouvez également trouver les</td>
</tr>
</tbody>
</table>
Scénario : Écrire des données d’une table d’une base de données dans un fichier LDIF

Ce scénario décrit un Job chargeant des données dans une table de base de données, extrayant les données de la table et écrivant ces données dans un nouveau fichier LDIF.

Construire le Job

Procédure
1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leurs noms dans l’espace de modélisation graphique ou en les déposant depuis la Palette : un tFixedFlowInput, un tMysqlOutput, un tMysqlInput et un tFileOutputLDIF.
2. Reliez le tFixedFlowInput au tMysqlOutput à l’aide d’un lien Row > Main.
3. Connectez le tMysqlInput au tFileOutputLDIF à l’aide d’un lien Row > Main.

Configurer les composants

Charger les données dans une table de base de données

Procédure
1. Double-cliquez sur le tFixedFlowInput pour ouvrir sa vue Basic settings.
2. Cliquez sur le bouton [...] à côté du champ **Edit schema** et définissez le schéma dans la fenêtre qui s’ouvre. Ajoutez quatre colonnes, respectivement : *dn*, *id_owners*, *registration* et *make*, de type **String**.

3. Cliquez sur **OK** pour fermer l’éditeur de schéma et acceptez la propagation proposée par la boîte de dialogue.

4. Dans la zone **Mode**, sélectionnez **Use Inline Content(delimited file)** et, dans le champ **Content** affiché, saisissez les données d’entrée suivantes :

   24;24;5382 KC 94;Volkswagen
   32;32;5981 0E 79;Honda
   35;35;3129 VH 61;Volkswagen

5. Double-cliquez sur le tMysqlOutput pour ouvrir sa vue **Basic settings**.
6. Renseignez les champs **Host**, **Port**, **Database**, **Username** et **Password** avec vos informations de connexion à la base de données MySQL.

7. Dans le champ **Table**, saisissez le nom de la table dans laquelle écrire les données. Dans cet exemple, saisissez `ldifdata`.

8. Sélectionnez **Drop table if exists and create** dans la liste **Action on table**.

**Extrait de données de la table de base de données et les écrire dans un fichier LDIF**

**Procédure**

1. Double-cliquez sur le *tMysqlInput* pour ouvrir sa vue **Basic settings**.
2. Renseignez les champs **Host**, **Port**, **Database**, **Username** et **Password** avec vos informations de connexion à la base de données MySQL.

3. Cliquez sur le bouton [...] à coté du champ **Edit schema** et, dans la fenêtre qui s'ouvre, définissez le schéma en ajoutant quatre colonnes, respectivement : *dn*, *id_owners*, *registration* et *make*, toutes de type **String**.

4. Dans le champ **Table Name**, saisissez le nom de la table de laquelle lire les données. Dans cet exemple, saisissez *ldifdata*.

5. Cliquez sur le bouton **Guess Query** pour renseigner le champ **Query** avec la requête auto-générée.

6. Double-cliquez sur le **tFileOutputLDIF** pour ouvrir sa vue **Basic settings**.

   ![tFileOutputLDIF](image)

   **Basic settings**

    - **File Name**: "E:/out.ldif"
    - **Wrap**: 78
    - **Change type**: Add
    - **Multi-Values**:
      - *dn* : ""
      - *id_owners* : ""
      - *registration* : ""
      - *make* : ""

7. Dans le champ **File Name**, parcourez votre système ou saisissez le chemin d'accès au fichier LDIF à générer. Dans cet exemple, saisissez *E:/out.ldif*.

8. Sélectionnez l'opération **Add** dans la liste **Change type**.

9. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** afin de l'exécuter.
Le fichier LDIF créé comporte les données de la table de la base de données. Le Changetype de ces entrées est défini à *addition*.
### tFileOutputMSDelimited

Ce composant crée un fichier délimité multi-structure complexe à l’aide de la structure des données (schémas) venant des différents flux d’entrée de type **Row**.

### Propriétés du tFileOutputMSDelimited Standard

Ces propriétés sont utilisées pour configurer le tFileOutputMSDelimited s’exécutant dans le framework de Jobs Standard.

Le composant tFileOutputMSDelimited Standard appartient à la famille **File**.

Le composant de ce framework est toujours disponible.

#### Basic settings

<table>
<thead>
<tr>
<th><strong>File Name</strong></th>
<th>Chemin d’accès et nom du fichier de sortie, et/ou variable à utiliser. Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chaîne (ex : &quot;\n&quot; sous Unix) séparant les lignes.</td>
<td></td>
</tr>
<tr>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
<td></td>
</tr>
<tr>
<td>Cochez cette case attribuer un séparateur différent à chacun des schémas, à partir de la colonne Field separator du champ Schemas.</td>
<td></td>
</tr>
</tbody>
</table>

#### Advanced settings

<table>
<thead>
<tr>
<th><strong>Advanced separator (for numbers)</strong></th>
<th>Cochez cette option pour modifier les séparateurs utilisés pour les nombres : Thousands separator : définissez le séparateur utilisé pour les milliers. Decimal separator : définissez le séparateur utilisé pour les décimaux.</th>
</tr>
</thead>
</table>
CSV options
Cochez cette case pour prendre en compte les paramètres spécifiques aux fichiers CSV, notamment la manière de protéger les caractères dans le champ Escape char et le type de guillemet dans le champ Text enclosure.

Create directory if not exists
Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier délimité de sortie s'il n'existe pas déjà.

Encoding
 Sélectionnez l'encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

Don't generate empty file
Cochez cette case pour annuler la génération du fichier si celui-ci est vide.

tStatCatcher Statistics
Cochez cette case pour collecter les métdonnées de process du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.

Global Variables

Global Variables

Global Variables

Global Variables

Global Variables

Global Variables

Global Variables

Global Variables

Global Variables

NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d'utilisation
Utilisez ce composant pour écrire un fichier délimité multi structure et séparer les champs à l'aide d'un séparateur de champ.

Limitation
Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton Install dans l'onglet Component. Vous pouvez également trouver les
Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center ([https://help.talend.com](https://help.talend.com)).

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tFileOutputMSPositional**

Ce composant crée un fichier multi-structure complexe à l’aide de la structure des données (schémas) venant des différents flux d’entrée de type **Row**.

**Propriétés du tFileOutputMSPositional Standard**

Ces propriétés sont utilisées pour configurer le tFileOutputMSPositional s’exécutant dans le framework de Jobs Standard.

Le composant tFileOutputMSPositional Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Row separator</td>
<td>Chaîne (ex : &quot;\n&quot; sous Unix) séparant les lignes.</td>
</tr>
</tbody>
</table>

**Schemas**

La table est automatiquement chargée avec des schémas résultant des différents flux d’entrée connectés au composant **tFileOutputMSPositional**. Renseignez les relations de dépendance entre les différents schémas :

- **Parent row** : Saisissez le nom du flux père (en fonction du nom du lien Row qui transfère les données).
- **Parent key column** : Saisissez le nom de la colonne clé du flux père.
- **Key column** : Saisissez le nom de la colonne clé pour la ligne sélectionnée.
- **Pattern** : Saisissez le modèle (pattern) qui permet de positionner les séparateurs de champ pour chaque ligne d’entrée.
- **Padding char** : Saisissez entre guillemets le caractère de remplissage à utiliser. Le caractère par défaut est un espace.
- **Alignment** : Sélectionnez le paramètre d’alignement approprié.

### Advanced settings

**Advanced separator (for numbers)**

Cochez cette option pour modifier les séparateurs utilisés pour les nombres :

- **Thousands separator** : définissez le séparateur utilisé pour les milliers.
### Decimal separator
Définissez le séparateur utilisé pour les décimaux.

### Create directory if not exists
Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier délimité de sortie s’il n’existe pas déjà.

### Encoding
Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

### tStatCatcher Statistics
Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

## Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable Flow et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable Flow et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_UNKOWN_HEADER_TYPES : nombre de lignes ayant un type d’en-tête inconnu. Cette variable est une variable Flow et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_PARSE_ERRORS : nombre de lignes ayant des erreurs de parsage. Cette variable est une variable Flow et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

## Utilisation

### Règle d’utilisation
Utilisez ce composant pour écrire un fichier positionnel multi structure et séparer les champs à l’aide d’un séparateur de position.
Scénario associé

Aucun scénario n'est disponible pour la version Standard de ce composant.
tFileOutputMSXML

Ce composant crée un fichier XML multi-structure complexe à l'aide de la structure des données (schémas) venant des différents flux d’entrée de type Row.

Propriétés du tFileOutputMSXML Standard

Ces propriétés sont utilisées pour configurer le tFileOutputMSXML s’exécutant dans le framework de Jobs Standard.
Le composant tFileOutputMSXML Standard appartient à la famille File.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure XML tree</td>
<td>Ouvre l’interface dédiée pour vous aider à mettre en place le mapping XML. Pour plus de détails sur l’interface, voir Définir un arbre XML Multischéma à la page 1203.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Create directory if not exists</th>
<th>Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier délimité de sortie s’il n’existe pas déjà.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced separator (for numbers)</td>
<td>Cochez cette option pour modifier les séparateurs utilisés pour les nombres : Thouands separator : définissez le séparateur utilisé pour les milliers. Decimal separator : définissez le séparateur utilisé pour les décimaux.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</td>
</tr>
<tr>
<td>Don’t generate empty file</td>
<td>Cochez cette case pour annuler la génération du fichier si celui-ci est vide.</td>
</tr>
<tr>
<td>Trim the whitespace characters</td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de colonnes.</td>
</tr>
<tr>
<td>Escape text</td>
<td>Cochez cette case pour échapper les caractères spéciaux.</td>
</tr>
</tbody>
</table>
Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

### Définir un arbre XML Multischéma

Double-cliquez sur le composant **tFileOutputMSXML** pour ouvrir l’interface dédiée ou cliquez sur le bouton [...] de l’onglet **Basic settings** de la vue **Component**.
La liste déroulante de la zone **Linker source**, à gauche de l’interface, contient tous les schémas d’entrée qui peuvent être ajoutés au fichier XML multischéma de sortie (à la condition que plusieurs flux d’entrée soient connectés au composant **tFileOutputMSXML**). 

Dans le champ **Schema List**, on retrouve la liste de toutes les colonnes récupérées du flux d’entrée de la sélection.

Sur la droite de l’interface, définissez la structure XML que vous souhaitez obtenir dans le fichier XML en sortie.

Vous pouvez la créer manuellement ou tout simplement importer la structure XML. Puis importez les colonnes de chaque schéma d’entrée de la sélection dans l’élément de l’arbre XML correspondant.

**Importer un arbre XML**

Le meilleur moyen de renseigner l’arbre XML est d’importer un fichier XML bien formé.

**Procédure**

1. Renommez la balise racine qui s’affiche par défaut dans le panneau **XML tree**, en cliquant sur celle-ci.
2. Dans la colonne **XML Tree**, cliquez-droit sur le champ **root tag** pour afficher le menu contextuel.
3. Dans le menu, sélectionnez **Import XML tree**.
4. Sélectionnez le fichier à importer et cliquez sur **OK**.
   - Vous pouvez importer la structure XML d’un fichier au format XML, XSD et DTD.
   - Lors de l’import d’une structure XML d’un fichier XSD, vous pouvez choisir un élément comme la racine de votre arbre XML.

La colonne **XML Tree** est donc automatiquement renseignée avec les éléments.
5. Si vous devez ajouter ou supprimer un élément ou un sous-élément, cliquez-droit sur l'élément correspondant, dans l’arborescence, pour afficher le menu contextuel.


**Créer manuellement l’arbre XML**

Si vous ne possédez pas de structure XML déjà définie, vous pouvez la créer manuellement.

**Procédure**

1. Dans la colonne **XML Tree**, cliquez une fois sur le champ **root tag** pour le renommer.
2. Cliquez-droit sur ce champ pour afficher le menu contextuel.
3. Dans le menu, sélectionnez **Add-sub-element** pour créer le premier élément de la structure.
4. Si vous devez ajouter un attribut ou un élément fils à un élément, ou si vous devez supprimer un élément, cliquez-droit à gauche du nom de l’élément correspondant, pour afficher le menu contextuel.
5. Dans le menu, sélectionnez l’option adéquate parmi les suivantes : **Add sub-element**, **Add attribute**, **Add namespace** ou **Delete**.

**Mapping de données XML à partir de sources multischéma**

Une fois votre arbre XML créé, sélectionnez le premier schéma à mapper. Vous pouvez alimenter chaque élément ou sous-élément XML avec les colonnes du flux d’entrée dans la colonne **Related Column**.

**Procédure**

1. Cliquez sur l’une des entrées de la colonne **Schema column name**.
2. Glissez-la dans le sous-élément correspondant à droite.
3. Relâchez-la pour que le mapping soit effectif.

Une flèche bleue apparaît pour illustrer ce mapping. S’il est disponible, utilisez le bouton **Auto-Map** située en bas à gauche de l’interface, il permet d’effectuer cette opération automatiquement.

4. Si vous devez déconnecter n’importe quel mapping de n’importe quel élément de l’arbre XML, sélectionnez l’élément de l’arbre XML que vous souhaitez déconnecter de sa source respective.
5. Cliquez-droit à gauche de son nom pour afficher le menu contextuel.
6. Sélectionnez **Disconnect linker**.
   La flèche bleue disparaît.

**Définir le statut du nœud**

Définir l’arbre XML et le transfert de données ne suffit pas. Vous devez aussi définir les éléments sur lesquels la boucle est effectuée **pour chacune des sources de la sélection** et, si nécessaire, l’élément à partir duquel le regroupement est effectué.

**Définir un élément Boucle**

L’élément Boucle permet de définir l’élément objet d’une itération. L’élément Boucle est généralement un générateur de lignes.

**Pourquoi et quand exécuter cette tâche**

Pour définir un élément comme un élément Boucle :

**Procédure**

1. Sélectionnez l’élément adéquat dans l’arbre XML.
2. Cliquez-droit à gauche du nom de l’élément pour afficher le menu contextuel.
3. Sélectionnez l’option **Set as Loop Element**.

**Résultats**

La colonne **Node Status** affiche le nouveau statut sélectionné.

**Remarque** :

Il ne peut y avoir qu’un seul élément Boucle à la fois.

**Définir un élément Group**

L’élément Group est optionnel, il représente un élément constant sur lequel est effectuée une fonction Groupby. L’élément Group ne peut être défini que si un élément Boucle a été préalablement défini.

**Pourquoi et quand exécuter cette tâche**

Lorsque vous utilisez un élément Group, les lignes doivent être triées pour pouvoir être regroupées par le nœud sélectionné.

Pour définir un élément comme élément Group :

**Procédure**

1. Sélectionnez l’élément adéquat dans l’arbre XML.
2. Cliquez-droit à gauche du nom de l’élément pour afficher le menu contextuel.
3. Sélectionnez l’option **Set as Group Element**.
Résultats

La colonne **Node Status** affiche le nouveau statut sélectionné et tout autre statut de regroupement est automatiquement défini, si nécessaire.

Une fois le mapping terminé, cliquez sur **OK** pour valider les paramètres et continuez la configuration de votre Job.

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tFileOutputPositional

Ce composant écrit un fichier ligne par ligne, en se basant sur la longueur et le format des champ ou colonnes d’une ligne.

Propriétés du tFileOutputPositional Standard

Ces propriétés sont utilisées pour configurer le tFileOutputPositional s’exécutant dans le framework de Jobs Standard.

Le composant tFileOutputPositional Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-In : propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
</tbody>
</table>

Use Output Stream

Cochez la case pour traiter le flux de données qui vous intéresse. Une fois cochée, le champ Output Stream s’affiche et vous pouvez saisir le flux de données souhaité.

Le flux de données à traiter doit être ajouté au flux afin que ce composant récupère ces données via la variable représentative correspondante.

Cette variable peut être prédéfinie dans votre Studio ou fournie par le contexte ou les composants utilisé(s) avec ce composant. Sinon, vous pouvez la définir manuellement et l’utiliser selon votre Job, par exemple à l’aide d’un tJava ou d’un tJavaFlex.

Afin d’éviter les désagréments de la saisie, vous pouvez sélectionner la variable qui vous intéresse dans la liste d’autocomplétion (Ctrl+Espace) afin de remplir le champ, si cette variable a été correctement définie.

Pour plus d’informations concernant l’utilisation d’un flux, consultez Scénario 2 : Lire les données d’un fichier distant en mode stream à la page 1072.

File name

Chemien’accès et nom du fichier de sortie et/ou variable à utiliser.

Ce champ est indisponible lorsque vous avez coché la case Use Output Stream.

Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend.

Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés.
au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Ce composant doit fonctionner avec un **tSetDynamicSchema** afin de tirer parti de la fonctionnalité de schéma dynamique.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.
- **Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

**Row separator**

Saisissez le séparateur utilisé pour identifier la fin d’une ligne.

**Append**

Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.

**Include header**

Cochez cette case pour tenir compte de l’en-tête dans le fichier de sortie.

**Compress as zip file**

Cochez cette case pour compresser en zip le fichier de sortie.

**Formats**

Personnalisez le format des données du fichier positionnel et renseignez les colonnes du tableau Formats.

- **Column** : Sélectionnez la colonne que vous souhaitez personnaliser.
- **Size** : Saisissez la taille correspondant à la colonne.
- **Padding char** : Saisissez entre guillemets le caractère de remplissage à utiliser. Le caractère par défaut est un espace.
- **Alignment** : Sélectionnez le paramètre d’alignement approprié.
| Keep | Si les données contenues dans la colonne ou dans le champ sont trop longues, sélectionnez la partie que vous souhaitez garder. |

**Advanced settings**

| **Advanced separator (for numbers)** | Cochez cette case pour modifier le séparateur utilisé pour les nombres. Par défaut, le séparateur des milliers est une virgule (,) et le séparateur décimal est un point (.).  
**Thousands separator** : définissez le séparateur utilisé pour les milliers.  
**Decimal separator** : définissez le séparateur utilisé pour les décimaux. |
| **Use byte length as the cardinality** | Cochez cette case pour permettre la prise en charge des caractères à deux octets dans ce composant. Pour cette fonction, JDK 1.6 est requis. |
| **Create directory if not exists** | Cochez cette case pour créer un nouveau dossier si votre dossier de destination n’existe pas. |
| **Custom the flush buffer size** | Cochez cette case pour personnaliser le nombre de lignes à écrire avant de vider le tampon.  
**Row Number** : Nombre de lignes à écrire avant de vider le tampon. |
| **Output in row mode** | Cochez cette case pour que l’écriture se fasse en mode ligne. |
| **Encoding** | Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données. |
| **Don’t generate empty file** | Cochez cette case pour empêcher la génération d’un fichier vide. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant. |

**Global Variables**

| **Global Variables** | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. |

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Utilisez ce composant pour lire un fichier et séparer les champs à l'aide du séparateur spécifié.</th>
</tr>
</thead>
</table>
| Dynamic settings   | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un **Studio Talend** indépendant.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez  **Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Scénario associé

Pour un scénario associé, consultez **Scénario : Transformer en fichier Regex en Positional** à la page 1144.

Pour un scénario relatif à l’utilisation de l’option **Use Output Stream**, consultez **Scénario 2 : Utiliser un flux de sortie pour sauvegarder des données dans un fichier local** à la page 1177.
tFileOutputProperties

Ce composant écrit un fichier de configuration de type .ini ou .properties contenant des données texte organisées sur le modèle clé=valeur.

Propriétés du tFileOutputProperties Standard

Ces propriétés sont utilisées pour configurer le tFileOutputProperties s’exécutant dans le framework de Jobs Standard.

Le composant tFileOutputProperties Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule. Il se compose de deux colonnes, Key et Value, correspondant au nom et à la valeur du paramètre à copier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>File format</td>
<td>Sélectionnez le format de votre fichier .properties ou .ini.</td>
</tr>
<tr>
<td>.properties : fichier de configuration dont les données sont disposées sur deux lignes et structurées de la manière suivante clé=valeur.</td>
<td></td>
</tr>
<tr>
<td>.ini : fichier de configuration dont les données sont disposées sur deux lignes, structurées de la manière suivante clé=valeur. et regroupées en sections..</td>
<td></td>
</tr>
<tr>
<td>Section Name</td>
<td>Saisissez le nom de la section sur laquelle effectuer l’opération d’itération.</td>
</tr>
<tr>
<td>Append</td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
</tbody>
</table>

Advanced settings

| Encoding | Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données. |
**tStatCatcher Statistics**

Cochez cette case pour collecter les métdonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

---

**Global Variables**

**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>Nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option. Une variable <em>Flow</em> fonctionne durant l’exécution d’un composant. Une variable <em>After</em> fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <em>Ctrl+Espace</em> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

---

**Utilisation**

**Règle d’utilisation**

Utilisez ce composant pour écrire un fichier dont les données sont organisées selon la structure clé=valeur.

---

**Scénario associé**

Pour un scénario associé, consultez *Scénario : Lire et mapper la clé et les valeurs de fichiers properties et alimenter un glossaire* à la page 1135 de la section *tFileInputProperties* à la page 1134.
tFileOutputRaw

Ce composant fournit des données provenant d'un autre composant en une seule colonne de sortie.

Propriétés du tFileOutputRaw Standard

Ces propriétés sont utilisées pour configurer le tFileOutputRaw s'exécutant dans le framework de Jobs Standard.
Le composant tFileOutputRaw Standard appartient à la famille File.
Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td>Filename</td>
<td>Nom et chemin d'accès au fichier d'entrée à traiter. Vous pouvez saisir le chemin manuellement entre guillemets doubles ou parcourir votre système en cliquant sur le bouton [...].</td>
</tr>
<tr>
<td>Encoding</td>
<td>Si cette sortie est une chaîne de caractères, sélectionnez le type d'encodage dans la liste ou sélectionnez Custom et définissez-le manuellement.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les...</td>
</tr>
</tbody>
</table>
lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject.
Pour capturer l’exception FileNotFoundException, vous devez cocher cette case.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | FILENAME_PATH : chemin du fichier d’entrée. Cette variable est une variable After et retourne une chaîne de caractères.  
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.  
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| Règle d’utilisation | Le composant tFileOutputRaw est utilisé pour recevoir des données provenant d’une source de données fournissant ces données en une seule colonne. |
tFileOutputXML

Ce composant écrit un fichier XML avec des valeurs de données séparées selon un schéma spécifié.

Propriétés du tFileOutputXML Standard

Ces propriétés sont utilisées pour configurer le tFileOutputXML s’exécutant dans le framework de Jobs Standard.

Le composant tFileOutputXML Standard appartient aux familles File et XML.

Le composant de ce framework est toujours disponible.

Basic settings

| File name | Chemin d’accès et nom du fichier de sortie, et/ou variable à utiliser.  
Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend. |
|-----------------|---------------------------------------------------------------|
| Incoming record is a document | Cochez cette case si les données du composant précédent sont au format XML.  
Lorsque cette case est cochée, une liste **Column list** apparaît et vous permet de sélectionner une colonne de type **Document** dans le schéma contenant les données.  
Le champ **Row tag** disparaît.  
Lorsque cette case est cochée, dans la vue **Advanced settings**, seules les cases **Create directory if not exists**, **Don’t generate empty file**, **Trim data**, **tStatCatcher Statistics** et la liste **Encoding** sont disponibles. |
| Row tag | Spécifiez la balise entourant les données et la structure de chaque ligne. |
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez |
### propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


### Sync columns

Cliquez sur ce bouton pour synchroniser le schéma de sortie avec le schéma d'entrée. La fonction Sync ne s'affiche que si une connexion de type Row est liée au composant d'entrée.

### Advanced settings

#### Split output in several files

Si le fichier XML en sortie est volumineux, vous pouvez scinder ce fichier en plusieurs, chacun d'entre eux contenant un nombre spécifié de lignes.

**Rows in each output file** : Spécifiez le nombre de lignes de chaque fichier de sortie.

#### Create directory if not exists

Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s'il n'existe pas déjà.

#### Root tags

Spécifiez une ou plusieurs balise(s) racine pour entourer la structure du fichier de sortie ainsi que les données. La balise racine par défaut est *root*.

#### Output format

Définissez le format de sortie.

- **Column** : la colonne récupérée du schéma d’entrée.
- **As attribute** : cochez la case des colonnes à utiliser comme attribut de l’élément parent dans la sortie XML.

**Remarque** :

Si la même colonne est sélectionnée dans la table Output format en tant qu’attribut et dans le paramètre Use dynamic grouping pour le regroupement dynamique, seul le paramètre de regroupement dynamique est pris en compte pour cette colonne.

**Use schema column name** : par défaut, cette case est cochée pour toutes les colonnes, afin que les libellés des colonnes du schéma d’entrée soient utilisés comme balises pour entourer les données. Si vous souhaitez utiliser une balise différente de celle du schéma d’entrée pour une colonne, décochez cette case pour cette colonne et spécifiez un libellé de balise entre guillemets, dans le champ Label.
| Use dynamic grouping | Cochez cette case si vous souhaitez regrouper dynamiquement les colonnes de sortie. Cliquez sur le bouton [+ ] pour ajouter un critère de regroupement dans la table Group by.

**Column** : sélectionnez une colonne à utiliser comme élément pour entourer les lignes de sortie groupées.

**Attribute label** : saisissez entre guillemets doubles un attribut de libellé pour l’élément entourant le groupe. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom the flush buffer size</td>
<td>Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ Row number, saisissez le nombre de lignes après lesquelles la mémoire est à nouveau libérée.</td>
</tr>
</tbody>
</table>
| Advanced separator (for numbers) | Cochez cette option pour modifier les séparateurs utilisés pour les nombres :

**Thousands separator** : définissez le séparateur utilisé pour les milliers.

**Decimal separator** : définissez le séparateur utilisé pour les décimaux. |
| Encoding | Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données. |
| Don’t generate empty file | Cochez cette case pour ne pas générer de fichier vide. |
| Trim data | Cochez cette case pour supprimer les espaces en début et fin du texte et fusionner les différents espaces consécutifs en un, dans le texte. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Utilisez ce composant pour écrire dans un fichier XML des données récupérées d’autres composants à l’aide d’une connexion de type Row. |

**Scénarios associés**

Pour des scénarios utilisant le composant **tFileOutputXML**, consultez Transformer un fichier positionnel en fichier XML à la page 1129 et Scénario 2 : Utiliser un message SOAP depuis un fichier XML pour obtenir le nom d’un pays et le sauvegarder dans un fichier XML à la page 3782.
tFileProperties

Ce composant crée un flux monoligne affichant les propriétés principales du fichier traité.

Propriétés du tFileProperties Standard

Ces propriétés sont utilisées pour configurer le tFileProperties s’exécutant dans le framework de Jobs Standard.
Le composant tFileProperties Standard appartient à la famille File.
Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit les champs qui sont traités et passés au composant suivant. Le schéma de ce composant est en lecture seule. Il décrit les propriétés principales du fichier spécifié. Vous pouvez cliquer sur le bouton [...] à côté de Edit schema pour voir le schéma prédéfini qui contient les champs suivants :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• abs_path : le chemin absolu du fichier.</td>
<td></td>
</tr>
<tr>
<td>• dirname : le répertoire du fichier.</td>
<td></td>
</tr>
<tr>
<td>• basename : le nom du fichier.</td>
<td></td>
</tr>
<tr>
<td>• mode_string : le mode d’accès du fichier, r et w pour les permissions de lecture et d’écriture respectivement.</td>
<td></td>
</tr>
<tr>
<td>• size : la taille du fichier en octets.</td>
<td></td>
</tr>
<tr>
<td>• mtime : le timestamp indiquant quand le fichier a été modifié pour la dernière fois, en millisecondes qui se sont écoulées depuis l’époque UNIX (1er janvier 1970, 00:00:00 UTC).</td>
<td></td>
</tr>
<tr>
<td>• mtime_string : la date et l’heure auxquelles le fichier a été modifié pour la dernière fois.</td>
<td></td>
</tr>
</tbody>
</table>


| Calculate MD5 Hash | Cochez cette case pour vérifier le MD5 du fichier téléchargé. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
Global Variables


Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en standalone.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connections</td>
<td>Liens de sortie (de ce composant à un autre) :</td>
</tr>
<tr>
<td></td>
<td>Row : Main, Iterate.</td>
</tr>
<tr>
<td></td>
<td>Liens d’entrée (d’un autre composant à celui-ci) :</td>
</tr>
<tr>
<td></td>
<td>Row : Iterate.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les liens, consultez le Guide utilisateur de Studio Talend.</td>
</tr>
</tbody>
</table>

Scénario : Afficher les propriétés d’un fichier traité

Ce scénario décrit un Job très simple permettant d’afficher les propriétés d’un fichier spécifié.

Procédure

Procédure

1. A partir de la Palette, cliquez-déposez les composants tFileProperties et tLogRow dans l’éditeur graphique.
2. Cliquez-droit sur le tFileProperties et connectez-le au tLogRow à l’aide d’un lien de type Row Main.
3. Dans l'éditeur graphique, sélectionnez le composant **tFileProperties**.

4. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tFileProperties**.

5. Configurez le champ **Schema** en mode **Built-In**.

6. Cliquez sur le bouton [...] du champ **Edit schema** si vous voulez voir les colonnes en lecture seule.

7. Dans le champ **File**, saisissez le chemin d'accès ou parcourez vos dossiers jusqu'au fichier dont vous voulez afficher les propriétés.

8. Dans l'éditeur graphique, sélectionnez le **tLogRow** et cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**). Pour plus d'informations, consultez **tLogRow** à la page 2105.

9. Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

```
Starting job File_Properties at 14:57 26/08/2008

key | value
---|---
abs_path | C:\test5.txt
dirname | C:\
bsename | test5.txt
ncode_string | rv
size | 06
mtime | 1219574736421
ntime_string | Mon Aug 25 16:32:16 CEST 2008

Job File_Properties ended at 14:57 26/08/2008. [exit code=0]
```

**Résultats**

Les propriétés du fichier défini s'affichent dans la console **Run**.
tFileRowCount

Ce composant ouvre un fichier et le lit ligne par ligne afin de déterminer le nombre de lignes qu’il contient.

Propriétés du tFileRowCount Standard

Ces propriétés sont utilisées pour configurer le tFileRowCount s’exécutant dans le framework de Jobs Standard.
Le composant tFileRowCount Standard appartient à la famille File.
Le composant de ce framework est toujours disponible.

Basic settings

| File Name                        | Chemin d’accès et nom du fichier à traiter, et/ou variable à utiliser.  
|                                 | Pour plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend. |
| Row separator                   | Chaîne (ex : "\n" sous Unix) séparant les lignes. |
| Ignore empty rows               | Cochez cette case pour que les lignes vides ne soient pas prises en compte. |
| Encoding                        | Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données. |

Advanced settings

| tStatCatcher Statistics         | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables

| Global Variables                | COUNT : nombre de lignes dans un fichier. Cette variable est une variable Flow et retourne un nombre entier.  
|                                 | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.  
|                                 | Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |
|                                 | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette |
### Utilisation

#### Règle d’utilisation

Le **tFileRowCount** est un composant standalone, et doit être utilisé avec un lien de type **OnSubjobOk** vers un **tJava**.

#### Connections

Liens de sortie (de ce composant à un autre) :
- **Row** : Main.
- **Trigger** : OnSubjobOk, OnSubjobError, Run if.

Liens d’entrée (d’un autre composant à celui-ci) :
- **Row** : Main, Reject, Iterate.

Pour plus d’informations concernant les liens, consultez la section relatives aux types de connexions, dans le **Guide utilisateur du Studio Talend**.

#### Limitation

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

### Scénario : Ecrire un fichier dans MySQL si le nombre d’enregistrements correspond à une valeur de référence

Dans ce scénario, le **tFileRowCount** compte le nombre d’enregistrements d’un fichier .txt et compare ce nombre à une valeur de référence grâce au **tJava**. Lorsque les deux valeurs correspondent, le fichier .txt est écrit dans une table MySQL.

Le fichier .txt contient deux enregistrements :

```
1;andy
2;mike
```
Relier les composants

Procédure

1. Déposez un `tFileRowCount`, un `tJava`, un `tFileInputDelimited` et un `tMysqlOutput` de la Palette dans l'espace de modélisation graphique.
2. Reliez le `tFileRowCount` au `tJava` à l'aide d'un lien `Trigger > OnSubjobOk`.
3. Reliez le `tJava` au `tFileInputDelimited` à l'aide d'un lien `Trigger > Run if`.
4. Reliez le `tFileInputDelimited` au `tMysqlOutput` à l'aide d'un lien `Row > Main`.

Configurer les composants

Procédure

1. Double-cliquez sur le `tFileRowCount` pour ouvrir sa vue `Basic settings`.

   ![tFileRowCount](image)

   1. Dans le champ **File Name**, saisissez le chemin complet vers le fichier .txt. Vous pouvez également cliquer sur le bouton `...` pour parcourir votre système jusqu’au fichier.
   2. Cochez ensuite la case **Ignore empty rows**.

2. Double-cliquez sur le `tJava` pour ouvrir sa vue `Basic settings`.

   ![tJava](image)

   ```java
   System.out.println(globalMap.get("tFileRowCount_1_COUNT"));
   ```
Dans la zone **Code**, saisissez la fonction permettant d’afficher le nombre de lignes du fichier :

```
System.out.println(globalMap.get("tFileRowCount_1_COUNT"));
```

4. **Cliquez sur le lien If pour ouvrir sa vue Basic settings.**

Dans la zone **Condition**, saisissez l’instruction permettant de déterminer si le nombre de lignes est égal à 2 :

```
((Integer)globalMap.get("tFileRowCount_1_COUNT")) == 2
```

Ce lien **If** signifie que si le nombre de lignes est égal à 2, les lignes du fichier .txt sont écrites dans MySQL.

5. **Double-cliquez sur le tFileInputDelimited pour ouvrir sa vue Basic settings.**

Dans le champ **File name/Stream**, saisissez le chemin complet vers le fichier .txt. Vous pouvez également cliquer sur le bouton afin de parcourir votre système jusqu’au fichier.

6. **Cliquez sur le bouton Edit schema afin d’ouvrir l’éditeur de schéma.**

![Diagramme des paramètres de tFileInputDelimited](image-url)
7. Cliquez deux fois sur le bouton afin d’ajouter deux colonnes, nommées respectivement id (de type Integer) et name (de type String).

8. Dans la boîte de dialogue qui s’affiche, cliquez sur le bouton Yes afin de propager le schéma au composant suivant.

![Propagate]


![tMysqlOutput_1]

10. Dans les champs Host et Port, saisissez les informations de connexion. Dans le champ Database, saisissez le nom de la base de données. Dans les champs Username et Password, saisissez les informations d’authentification. Dans le champ Table, saisissez le nom de la table, par exemple "staff". Dans la liste Action on table, sélectionnez Create a table if not exists. Dans la liste Action on data, sélectionnez Insert.

**Exécuter le Job**

**Procédure**

1. Appuyez sur Ctrl+S afin de sauvegarder votre Job.
2. Appuyez sur F6 pour exécuter le Job.
Comme montré ci-dessus, le Job a été exécuté avec succès et le nombre de lignes du fichier .txt a été affiché.

3. Ouvrez l’interface graphique de MySQL et ouvrez la table staff.

Comme montré ci-dessus, la table a été créée et deux enregistrements ont été insérés.
**tFileTouch**

Ce composant crée un fichier vide ou, si le fichier spécifié existe déjà, met à jour sa date de modification et de dernier accès et laisse son contenu inchangé.

**Propriétés du tFileTouch Standard**

Ces propriétés sont utilisées pour configurer le tFileTouch s'exécutant dans le framework de Jobs Standard.

Le composant tFileTouch Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>File Name</th>
<th>Chemin d'accès et nom du fichier à créer, et/ou variable à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create directory if not exists</td>
<td>Cochez cette case pour créer un nouveau dossier si votre dossier de destination n'existe pas.</td>
</tr>
</tbody>
</table>

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables


### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant peut être utilisé en stand alone.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connections</td>
<td>Liens de sortie (de ce composant à un autre) :</td>
</tr>
<tr>
<td>Row</td>
<td>Main.</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
</tr>
</tbody>
</table>

Liens d’entrée (d’un autre composant à celui-ci) :

<table>
<thead>
<tr>
<th>Row</th>
<th>Main, Reject, Iterate.</th>
</tr>
</thead>
</table>

Pour plus d’informations concernant les liens, consultez le Guide utilisateur du Studio Talend.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tFileUnarchive


Propriétés du tFileUnarchive Standard

Ces propriétés sont utilisées pour configurer le tFileUnarchive s’exécutant dans le framework de Jobs Standard.
Le composant tFileUnarchive Standard appartient à la famille File.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive file</td>
<td>Chemin d’accès au fichier archive.</td>
</tr>
<tr>
<td>Extraction Directory</td>
<td>Répertoire dans lequel le fichier dézippé sera placé.</td>
</tr>
<tr>
<td>Use archive file name as root directory</td>
<td>Cochez cette case pour créer un dossier du même nom que l’archive, s’il n’existe pas, dans le répertoire spécifié et extraire le(s) fichier(s) dézippé(s) dans ce dossier.</td>
</tr>
<tr>
<td>Check the integrity before unzip</td>
<td>Cochez cette case pour effectuer une vérification d’intégrité avant de dézipper l’archive.</td>
</tr>
<tr>
<td>Extract file paths</td>
<td>Cochez cette case pour reproduire la structure du chemin d’accès au fichier zippé dans l’archive.</td>
</tr>
<tr>
<td>Need a password</td>
<td>Cochez cette case et saisissez le mot de passe et la méthode de décryptage corrects si la décompression de l’archive est protégée par un mot de passe. Notez que l’archive cryptée doit être créée par le composant tFileArchive, sinon des messages d’erreur apparaîtront ou rien ne sera extrait, même si aucun message d’erreur n’est apparu. Decrypt method : sélectionnez dans la liste la méthode de décryptage Java Decrypt ou Zip4j Decrypt. Enter the password : saisissez le mot de passe de décryptage. Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>
Global Variables

| Global Variables | CURRENT_FILEPATH : nom du fichier courant ainsi que son chemin d'accès. Cette variable est une variable Flow et retourne une chaîne de caractères. |
| Global Variables | ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
| Global Variables | Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant. |
| Global Variables | Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
| Global Variables | Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en standalone mais il peut aussi être utilisé en composant de départ dans un Job via un lien Iterate. |
| Connections | Liens de sortie (de ce composant à un autre) :
Row : Iterate. |
| Connections | Liens d’entrée (d’un autre composant à celui-ci) :
Row : Iterate. |
| Connections | Pour plus d’informations concernant les liens, consultez la section relative aux types de connexions, dans le Guide utilisateur du Studio Talend. |

Limitation

| Limitation | Avertissement :
Scénario associé

Pour un exemple d’utilisation du composant *tFileUnarchive*, consultez *tFileCompare* à la page 1035.
**tFilterColumns**

Ce composant homogénéise des schémas, en organisant les colonnes ou en supprimant ou en ajoutant des colonnes.

**Propriétés du tFilterColumns Standard**

Ces propriétés sont utilisées pour configurer le tFilterColumns s'exécutant dans le framework de Jobs Standard.

Le composant tFilterColumns Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-In</strong></td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et il est stocké dans le <strong>Repository</strong>. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

| Global Variables | GLOBAL_VARIABLES | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. | GLOBAL_VARIABLES |
| | | NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. | |
| | | Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. | |
| | | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. | |
| | | Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. | |

**Utilisation**

| Règle d’utilisation | Ce composant n’est pas un composant de début (fond vert) et il nécessite un composant de sortie. | Règle d’utilisation |

**Scénario associé**

Pour plus d’informations concernant l’utilisation du composant tFilterColumns, consultez Scénario : Remplacement multiple et filtrage de colonnes à la page 3360 du composant tReplace.
**tFilterRow**

Ce composant filtre des lignes d’entrée en définissant une ou plusieurs condition(s) sur les colonnes sélectionnées.

**Propriétés du tFilterRow Standard**

Ces propriétés sont utilisées pour configurer le tFilterRow s’exécutant dans le framework de Jobs Standard.

Le composant tFilterRow Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Le schéma de ce composant est en Built-in seule. |
|----------------------|--------------------------------------------------------------------------------------------------|

| Logical operator used to combine conditions | Sélectionnez un opérateur logique afin de combiner des conditions simples et de combiner les résultats de filtrer des deux modes si des conditions avancées ont été définies.

**And** : retourne la valeur booléenne true si toutes les conditions sont vraies. Sinon, retourne false. Pour chacune des deux conditions combinées à l’aide de l’opérateur logique AND, la seconde condition est évaluée uniquement si la première est true.

**Or** : retourne la valeur booléenne true si une condition est vraie. Sinon, retourne la valeur false. Pour les deux conditions combinées à l’aide de l’opérateur logique OR, la seconde condition est évaluée uniquement si la première est false. |
|------------------------------------------|--------------------------------------------------------------------------------------------------|

| Conditions | Cliquez sur le bouton [+] pour ajouter autant de conditions simples que nécessaire. Selon l’opérateur logique sélectionné, les conditions sont évaluées les unes après les autres en ordre séquentiel pour chaque ligne. Lorsqu’elles sont évaluées, les conditions retournent la valeur booléenne true ou false.

**Input column** : Sélectionnez la colonne du schéma sur laquelle exécuter la fonction.

**Function** : Sélectionnez une fonction dans la liste.

**Operator** : Sélectionnez l’opérateur permettant de relier la colonne d’entrée et la valeur.

**Value** : Renseignez la valeur filtrée, entre guillemets si besoin est. |
|-------------------|--------------------------------------------------------------------------------------------------|
## Use advanced mode

Cochez cette case lorsque les opérations à effectuer ne peuvent aboutir via les fonctions standard proposées, par exemple différentes opérations logiques dans le même composant. Dans le champ texte, saisissez l’expression régulière comme demandé.

Si plusieurs conditions avancées sont définies, utilisez un opérateur logique entre deux conditions :

- **& &** (opérateur logique AND) : retourne la valeur booléenne *true* si les deux conditions sont vraies, sinon retourne *false*. La seconde condition est évaluée uniquement si la première condition est *true*.

- **|=** (opérateur logique OR) : retourne la valeur booléenne *true* si l’une des conditions est vraie, sinon, retourne *false*. La seconde condition est évaluée uniquement si la première est *false*.

## Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

## Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. |
| | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier. |
| | **NB_LINE_OK** : nombre de lignes correspondant au filtre. Cette variable est une variable *After* et retourne un entier. |
| | **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable *After* et retourne un entier. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

## Utilisation

| Règle d’utilisation | Ce composant n’est pas un composant de début (fond vert) et il nécessite un composant de sortie. |
Scénario 1 : Filtrer une liste de noms à l'aide de conditions simples

Le scénario suivant décrit un Job qui utilise des conditions simples pour filtrer une liste d'enregistrements. Le Job écrit deux tables de sortie : la première liste toutes les personnes de sexe masculin dont le nom de famille contient moins de neuf caractères et âgées de 10 à 80 ans. La seconde liste tous les enregistrements rejetés. Un message d'erreur s’affichera dans la table pour chaque enregistrement rejeté, afin d’expliquer pourquoi l’enregistrement a été rejeté.

Construire le Job

Procédure

3. Déposez de la Palette dans l’espace de modélisation graphique un nouveau tLogRow et renommez-le reject. Reliez le tFilterRow à ce tLogRow à l’aide d’un lien Row > Reject.
4. Renommez les autres composants afin de mieux identifier leur rôle au sein du Job.

Configurer les composants

Procédure

1. Dans l'espace graphique, cliquez sur le composant tFixedFlowInput afin d'afficher sa vue Basic settings et définir ses propriétés.
2. Cliquez sur le bouton [...] à côté du champ Edit schema afin de définir le schéma des données d’entrée. Dans cet exemple, le schéma se composant des quatre colonnes suivantes : LastName (type String), Gender (type String), Age (type Integer) et City (type String).
Cela fait, cliquez sur **OK** afin de valider les paramètres du schéma et fermer la boîte de dialogue. Une nouvelle boîte de dialogue s’ouvre et vous propose de propager le schéma. Cliquez sur **Yes**.

3. Dans les champs **Row separator** et **Field separator**, saisissez respectivement le séparateur de lignes et le séparateur de champs, si nécessaire. Dans cet exemple, utilisez les paramètres par défaut pour les deux champs, respectivement un retour chariot et un point-virgule.

4. Sélectionnez l’option **Use Inline Content(delimited file)** dans la zone **Mode** et saisissez les données d’entrée dans le champ **Content**.

Les données d’entrée utilisées dans cet exemple sont les suivantes :

<table>
<thead>
<tr>
<th>Nom</th>
<th>Genre</th>
<th>Age</th>
<th>Ville</th>
</tr>
</thead>
<tbody>
<tr>
<td>Van Buren</td>
<td>M</td>
<td>73</td>
<td>Chicago</td>
</tr>
<tr>
<td>Adams</td>
<td>M</td>
<td>40</td>
<td>Albany</td>
</tr>
<tr>
<td>Jefferson</td>
<td>F</td>
<td>66</td>
<td>New York</td>
</tr>
<tr>
<td>Adams</td>
<td>M</td>
<td>9</td>
<td>Albany</td>
</tr>
<tr>
<td>Jefferson</td>
<td>M</td>
<td>30</td>
<td>Chicago</td>
</tr>
<tr>
<td>Carter</td>
<td>F</td>
<td>26</td>
<td>Chicago</td>
</tr>
<tr>
<td>Harrison</td>
<td>M</td>
<td>40</td>
<td>New York</td>
</tr>
<tr>
<td>Roosevelt</td>
<td>F</td>
<td>15</td>
<td>Chicago</td>
</tr>
<tr>
<td>Monroe</td>
<td>M</td>
<td>8</td>
<td>Boston</td>
</tr>
</tbody>
</table>
5. Double-cliquez sur le composant **tFilterRow** pour afficher sa vue **Basic settings** et définir ses propriétés.

6. Dans le tableau **Conditions**, ajoutez quatre conditions et renseignez les paramètres de filtre.

7. Pour combiner les deux conditions, sélectionnez l’opérateur logique **And** dans la liste **Logical operator used to combine conditions**, pour accepter uniquement les enregistrement répondant aux deux conditions.

8. Dans la vue **Basic settings** des composants **tLogRow**, sélectionnez l’option **Table (print values in cells of a table)** dans la zone **Mode**.

**Exécuter le Job**

**Procédure**

Sauvegardez votre Job puis appuyez sur **F6** pour l’exécuter.
Résultats

<table>
<thead>
<tr>
<th>statistics</th>
<th>connected</th>
</tr>
</thead>
<tbody>
<tr>
<td>accepteds</td>
<td>------------</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LastName</td>
<td>Gender</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>Adams</td>
<td>M</td>
</tr>
<tr>
<td>Harrison</td>
<td>M</td>
</tr>
<tr>
<td>Arthur</td>
<td>M</td>
</tr>
<tr>
<td>Pierce</td>
<td>M</td>
</tr>
<tr>
<td>McKinley</td>
<td>M</td>
</tr>
<tr>
<td>Monroe</td>
<td>M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>statistics</th>
<th>disconnected</th>
</tr>
</thead>
<tbody>
<tr>
<td>rejecteds</td>
<td>---------------</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>LastName</td>
<td>Gender</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>Van Buren</td>
<td>M</td>
</tr>
<tr>
<td>Jefferson</td>
<td>F</td>
</tr>
<tr>
<td>Adams</td>
<td>M</td>
</tr>
<tr>
<td>Jefferson</td>
<td>M</td>
</tr>
<tr>
<td>Carter</td>
<td>F</td>
</tr>
<tr>
<td>Roosevelt</td>
<td>F</td>
</tr>
<tr>
<td>Monroe</td>
<td>M</td>
</tr>
<tr>
<td>Quincy</td>
<td>F</td>
</tr>
<tr>
<td>Coolidge</td>
<td>M</td>
</tr>
</tbody>
</table>

La première table liste les enregistrements des personnes de sexe masculin, âgées de 10 à 80 ans, dont le nom de famille comprend moins de 9 caractères. La seconde affiche tous les enregistrements ne correspondant pas aux conditions de filtre. À chaque enregistrement rejeté correspond un message d’erreur expliquant la raison du rejet.

Scénario 2 : Filtrer une liste de noms via différentes opérations logiques

Ce scénario se base sur le scénario précédent et filtre les données d’entrée afin que seuls les enregistrements des personnes de New York et Chicago soient acceptées. Sans modifier les paramètres de filtre du précédent scénario, des conditions avancées sont ajoutées pour permettre les opérations logiques OR et AND dans le même composant tFilterRow.

Procédure

1. Double-cliquez sur le composant tFilterRow pour afficher sa vue Basic settings.
2. Cochez la case **Use advanced mode** et saisissez l’expression suivante dans le champ **Advanced** :

```
input_row.City.equals("Chicago") || input_row.City.equals("New York")
```

Cette expression définit deux conditions sur la colonne *City* des données d’entrée afin de filtrer les enregistrements contenant les villes de Chicago et New York, et utilise l’opérateur logique OR pour accepter les enregistrements satisfaisant l’une des deux conditions.

3. Appuyez sur les touches **Ctrl+S** afin de sauvegarder le Job et appuyez sur la touche **F6** pour l’exécuter.

La liste des résultats du scénario précédent a été plus filtrée. Seuls les enregistrements contenant les villes de New York et Chicago sont acceptés.
tFirebirdClose

Ce composant permet de fermer une connexion à la base de données Firebird.

Propriétés du tFirebirdClose Standard

Ces propriétés sont utilisées pour configurer le tFirebirdClose s’exécutant dans le framework de Jobs Standard.
Le composant tFirebirdClose Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tFirebirdConnection dans la liste.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tFirebirdCommit**

Ce composant commite une transaction globale au lieu de commiter chaque ligne ou chaque lot de ligne et permet un gain de performance.

**Propriétés du tFirebirdCommit Standard**

Ces propriétés sont utilisées pour configurer le tFirebirdCommit s’exécutant dans le framework de Jobs Standard.

Le composant tFirebirdCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

ℹ️ **Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant <strong>tFirebirdConnection</strong> dans la liste.</td>
</tr>
</tbody>
</table>
| Close connection | Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.  

⚠️ **Avertissement :**  

Si vous utilisez un lien de type **Row > Main** pour relier le **tFireBirdCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne. |

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

1245
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants Firebird et notamment avec les composants tFirebirdConnection et tFirebirdRollback.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.  

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez le Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et le Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend. |

Scénario associé

Pour un scénario associé au composant tFirebirdCommit, consultez le Scénario : Insérer des données dans des tables mère/fille à la page 2620.
# tFirebirdConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

## Propriétés du tFirebirdConnection Standard

Ces propriétés sont utilisées pour configurer le tFirebirdConnection s’exécutant dans le framework de Jobs Standard.

Le composant tFirebirdConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

### Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

## Basic settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Table Schema</td>
<td>Nom du schéma.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Use or register a shared DB Connection | Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion |
partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.

### Advanced settings

| Auto Commit | Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit. Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec des composants Firebird, notamment les composants tFirebirdCommit et tFirebirdRollback. |
Scénario associé

Pour un scénario associé au composant **tFirebirdConnection**, consultez *Scénario : Insérer des données dans des tables mère/fille* à la page 2620.
tFirebirdInput

Ce composant exécute une requête en base de données sur une base Firebird selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

Propriétés du tFirebirdInput Standard

Ces propriétés sont utilisées pour configurer le tFirebirdInput s’exécutant dans le framework de Jobs Standard.

Le composant tFirebirdInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
</tbody>
</table>

**Built-in** : Propriétés utilisées ponctuellement.

**Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| Query type et Query | Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma. |
### Advanced settings

<table>
<thead>
<tr>
<th>Trim all the String/Char columns</th>
<th>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim column</td>
<td>Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>NB_LINE</strong> : nombre de lignes traitées. Cette variable est une variable <em>After</em> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>QUERY</strong> : requête traitée. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <em>Flow</em> fonctionne durant l’exécution d’un composant. Une variable <em>After</em> fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <em>Ctrl+Espace</em> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Firebird.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [*+] pour ajouter une ligne à la table. Dans le champ <em>Code</em>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>La table <em>Dynamic settings</em> est disponible uniquement lorsque la case <em>Use an existing connection</em> est cochée dans la vue <em>Basic settings</em>. Lorsqu’un paramètre</td>
</tr>
</tbody>
</table>

**Limitation**

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

**Scénarios associés**

Pour des scénarios associés, consultez :

Consultez également Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520.
tFirebirdOutput

Ce composant exécute l’action définie sur la table et/ou sur les données d’une table d’une base de données Firebird, en fonction du flux entrant provenant du composant précédent.
Le tFirebirdOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

Propriétés du tFirebirdOutput Standard

Ces propriétés sont utilisées pour configurer le tFirebirdOutput s’exécutant dans le framework de Jobs Standard.
Le composant tFirebirdOutput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Remarque :</td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
<td></td>
</tr>
</tbody>
</table>
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro de port d'écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d'authentification de l'utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Table          | Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois. |
| Action on table| Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :  
None : n'effectuer aucune opération de table.  
Drop and create the table : supprimer la table puis en créer une nouvelle.  
Create a table : créer une table qui n'existe pas encore.  
Create table if doesn't exist : créer la table si nécessaire.  
Drop a table if exists and create : supprimer la table si elle existe déjà, puis en créer une nouvelle.  
Clear a table : supprimer le contenu de la table. |
| Action on data | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :  
Insert : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.  
Update : Mettre à jour les entrées existantes.  
Insert or update : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.  
Update or insert : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré.  
Delete : Supprimer les entrées correspondantes au flux d'entrée.  
Avertissement : Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle... |
baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

<table>
<thead>
<tr>
<th><strong>Schema et Edit schema</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Built-In</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Repository</strong></th>
</tr>
</thead>
</table>
| Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.  

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets. 

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com). |

<table>
<thead>
<tr>
<th><strong>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</strong></th>
</tr>
</thead>
</table>
| • **View schema** : sélectionnez cette option afin de voir le schéma.  

• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
### Die on error
Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Commit every</strong></td>
<td>Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de transaction (cependant pas de rollback) et surtout une meilleure performance des exécutions.</td>
</tr>
<tr>
<td><strong>Additional Columns</strong></td>
<td>Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</td>
</tr>
</tbody>
</table>

#### Name
- **Saisissez le nom de la colonne à modifier ou à insérer.**

#### SQL expression
- **Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.**

#### Position
- ** Sélectionnez Before, Replace ou After, en fonction de l'action à effectuer sur la colonne de référence.**

#### Reference column
- **Saisissez une colonne de référence que le composant tFirebirdOutput peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.**

### Use field options
- Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.

### Enable debug mode
- Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.

### Support null in “SQL WHERE” statement
- Cochez cette case pour prendre en compte les valeurs Null d’une table de base de données.

#### Remarque:
- Assurez-vous que la case **Nullable** est bien cochée pour les colonnes du schéma correspondantes.

### Use Batch
- Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.

#### Remarque:
Cette case est disponible lorsque vous sélectionnez Insert, Update, ou Delete dans la liste Action on data.

**Batch Size**
Spécifiez le nombre d’enregistrements à traiter dans chaque lot.
Ce champ est disponible uniquement lorsque la case Use batch mode est cochée.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

- **NB_LINE** : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.
- **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.
- **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.
- **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.
- **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**
Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données Firebird. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer
### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Limitation

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

### Scénarios associés

Pour un scénario associé, consultez :

- **Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput** à la page 2667 du composant **tMysqlOutput**.
**tFirebirdRollback**

Ce composant annule la transaction commissée dans la base de données Firebird connectée.

**Propriétés du tFirebirdRollback Standard**

Ces propriétés sont utilisées pour configurer le tFirebirdRollback s’exécutant dans le framework de Jobs Standard.

Le composant tFirebirdRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant de connexion tFirebirdConnection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en association avec des composants Firebird, notamment avec le tFirebirdConnection et le tFirebirdCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez...</td>
</tr>
</tbody>
</table>
pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

Pour un scénario associé au **tFirebirdRollback**, consultez **Scénario : Annuler l’insertion de données dans des tables mère/fille** à la page 2623 du composant **tMysqlRollback**.
**tFirebirdRow**

Ce composant exécute la requête SQL déclarée sur la base de données Firebird spécifiée.

Selon la nature de la requête et de la base de données, tFirebirdRow agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Le tFirebirdRow est le composant spécifique à ce type de base de données. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

**Propriétés du tFirebirdRow Standard**

Ces propriétés sont utilisées pour configurer le tFirebirdRow s’exécutant dans le framework de Jobs Standard.

Le composant tFirebirdRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
</tbody>
</table>
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| **Username et Password** | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.  
**Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.  
**Repository** : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. |
| Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
• **View schema** : sélectionnez cette option afin de voir le schéma.  
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Nom de la table à traiter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.</td>
</tr>
<tr>
<td>Query</td>
<td>Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Rejects.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Propagate QUERY's recordset</th>
<th>Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="https://via.placeholder.com/15" alt="" /> Remarque :</td>
</tr>
<tr>
<td></td>
<td>Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type Object. Ce composant est généralement suivi du tParseRecordSet.</td>
</tr>
<tr>
<td></td>
<td><img src="https://via.placeholder.com/15" alt="" /> <strong>Parameter Index</strong> : Saisissez la position du paramètre dans l'instruction SQL.</td>
</tr>
<tr>
<td></td>
<td><img src="https://via.placeholder.com/15" alt="" /> <strong>Parameter Type</strong> : Saisissez le type du paramètre.</td>
</tr>
<tr>
<td></td>
<td><img src="https://via.placeholder.com/15" alt="" /> Remarque :</td>
</tr>
<tr>
<td></td>
<td>Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.</td>
</tr>
<tr>
<td>Commit every</td>
<td>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit</td>
</tr>
</tbody>
</table>
la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL. |

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir... |
de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Limitation


Scénarios associés

Pour un scénario associé, consultez :

- **Scénario : Combiner deux flux pour une sortie sélective** à la page 2706.
- **Procédure** du composant tDBSQLRow.
- **Scénario : Supprimer et re-générer un index de table MySQL** à la page 2700 du composant tMysqlRow.
**tFixedFlowInput**

Ce composant génère des données fixes à partir de variables internes.

**Propriétés du tFixedFlowInput Standard**

Ces propriétés sont utilisées pour configurer le tFixedFlowInput s’exécutant dans le framework de Jobs Standard.

Le composant tFixedFlowInput Standard appartient à la famille Misc.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend .</td>
<td></td>
</tr>
</tbody>
</table>
| **Mode** | Sélectionnez le mode, parmi les trois proposés, que vous souhaitez utiliser.
**Use Single Table** : Saisissez la donnée que vous souhaitez générer dans la colonne Value correspondante.
**Use Inline Table** : Ajoutez la ou les ligne(s) que vous souhaitez générer. |
**Use Inline Content** : Saisissez vos données à générer, séparées par les séparateurs préalablement définis dans les champs **Row** et **Field Separator**.

**Number of rows**
Saisissez le nombre de lignes de données à générer.

**Values**
Saisissez entre guillemets les valeurs correspondant aux colonnes définies dans la boîte de dialogue du schéma, via le bouton **Edit schema**.

**Advanced settings**

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

**Règle d’utilisation**
Ce composant peut être utilisé comme composant de début ou de milieu, il requiert donc un composant de sortie.

**Scénarios associés**

Pour des scénarios associés, consultez :

- **Scénario 2** : Mettre les données de sortie en mémoire tampon du serveur d’application Web à la page 427.

- **Scénario** : Itérer une table de base de données et lister le nom des colonnes de la table à la page 2611.

- **Scénario 1** : Filttrer une liste de noms à l’aide de conditions simples à la page 1238.
tFlowMeter

Ce composant compte le nombre de lignes traitées dans le flux défini, afin de récupérer ce nombre dans le tFlowMeterCatcher à des fins de log.

Propriétés du tFlowMeter Standard

Ces propriétés sont utilisées pour configurer le tFlowMeter s’exécutant dans le framework de Jobs Standard.
Le composant tFlowMeter Standard appartient à la famille Logs & Errors.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use input connection name as label</th>
<th>Cochez cette case pour réutiliser le nom donné au flux d’entrée Main Row comme libellé des données loggées.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>Sélectionnez le type de valeur des données mesurées :</td>
</tr>
<tr>
<td></td>
<td><strong>Absolute</strong> : le nombre actuel de lignes est loggé.</td>
</tr>
<tr>
<td></td>
<td><strong>Relative</strong> : un pourcentage (%) du nombre de lignes est loggé. Lorsque cette option est sélectionnée, la liste <strong>Connections List</strong> vous permet de sélectionner une connexion de référence.</td>
</tr>
<tr>
<td>Thresholds</td>
<td>Ajoutez un seuil pour observer les proportions en volumes mesurés. Vous pouvez décider que le flux doit être compris entre un nombre minimum et maximum de lignes, et si le flux est inférieur au seuil minimum, il y a un goulot d’étranglement.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette |
liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant ne peut être utilisé comme composant de début puisqu’il nécessite un flux d’entrée pour fonctionner. |

Si vous avez besoin d’utiliser les données de log, les statistiques et tout autre mesure des flux de données, consultez le Guide utilisateur du Studio Talend.

**Scénario associé**

Pour plus d’informations sur l’utilisation du tFlowMeter, consultez Scénario : Mesurer le flux d’un Job à la page 1272.
tFlowMeterCatcher

Ce composant fonctionne comme une fonction de log déclenchée par l'utilisation du composant tFlowMeter dans le Job.

Le composant tFlowMeterCatcher est basé sur le schéma défini. Il récupère les données du composant tFlowMeter et les transmet au composant de sortie.

Propriétés du tFlowMeterCatcher Standard

Ces propriétés sont utilisées pour configurer le tFlowMeterCatcher s'exécutant dans le framework de Jobs Standard.

Le composant tFlowMeterCatcher Standard appartient à la famille Logs & Errors.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit les champs à traiter et à passer au composant suivant. Pour ce composant, le schéma est en lecture seule puisqu'il regroupe les informations de log standard, notamment :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Moment</td>
<td>: Date et heure du traitement</td>
</tr>
<tr>
<td>Pid</td>
<td>: Identifiant de processus du Job.</td>
</tr>
<tr>
<td>Father_pid</td>
<td>: Identifiant de processus du Job parent, le cas échéant. Sinon, le Pid est dupliqué.</td>
</tr>
<tr>
<td>Root_pid</td>
<td>: Identifiant de processus du Job racine, le cas échéant. Sinon, le Pid est dupliqué.</td>
</tr>
<tr>
<td>System_pid</td>
<td>: Identifiant de processus généré par le système.</td>
</tr>
<tr>
<td>Project</td>
<td>: Nom du projet auquel appartient le Job.</td>
</tr>
<tr>
<td>Job</td>
<td>: Nom du Job en cours.</td>
</tr>
<tr>
<td>Job_repository_id</td>
<td>: Identifiant généré par l’application.</td>
</tr>
<tr>
<td>Job_version</td>
<td>: Numéro de version du Job en cours.</td>
</tr>
<tr>
<td>Context</td>
<td>: Nom du contexte en cours.</td>
</tr>
<tr>
<td>Origin</td>
<td>: Nom du composant, le cas échéant.</td>
</tr>
<tr>
<td>Label</td>
<td>: Libellé de la connexion Main Row précédent le composant tFlowMeter dans le Job et dont la volumétrie sera analysée.</td>
</tr>
<tr>
<td>Count</td>
<td>: Nombre de lignes en cours de traitement.</td>
</tr>
</tbody>
</table>
**Reference** : Nombre de lignes passant dans le lien Reference.

**Thresholds** : cette fonction ne s’utilise que si le mode Relative a été sélectionné dans le composant tFlowMeter.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Global Variables


### Utilisation

| Règle d’utilisation | Ce composant est le composant de début du Job secondaire qui se déclenche automatiquement à la fin du Job principal. |

| Limitation | L’utilisation de ce composant est liée à l’utilisation du composant tFlowMeter. Pour plus d’informations, consultez tFlowMeterCatcher à la page 1271. |

### Scénario : Mesurer le flux d’un Job

Le Job suivant permet de recueillir le nombre de lignes traitées. Les mesures sont prises deux fois, une fois après le composant d’entrée, c’est-à-dire avant qu’elles soient filtrées, et après leur filtrage, avant le composant de sortie.
• Cliquez et déposez les composants suivants : un **tMysqlInput**, deux **tFlowMeter**, un **tMap**, un **tLogRow**, un **tFlowMeterCatcher** et un **tFileOutputDelimited**.

• Connectez les composants du Job principal à l'aide de liens de type Row main et cliquez sur le libellé pour lui donner un nom plus parlant, par exemple *US_States* après le composant d’entrée et *filtered_states* après le **tMap**.

• Reliez aussi les composants **tFlowMeterCatcher** et **tFileOutputCSV** à l’aide d’une connexion de type Row main puisque les données sont transmises au composant de sortie.

• Dans la vue **Component** du composant **tMysqlInput**, sélectionnez l’option **Repository** dans le champ **Property Type**, si les propriétés de connexion à la base de données sont stockées dans une métadonnée du **Repository**. Sinon, sélectionnez l’option **Built-in** et paramétrez les propriétés et le schéma manuellement.

  - **Property Type**: **Repository**
  - **DB Version**: **Mysql 5**
  - **Use an existing connection**
  - **Host**: *localhost*  
  - **Database**: *tox*10*
  - **Username**: *root*
  - **Password**: ********
  - **Schema**: **Built-in**
  - **Table Name**: *states*
  - **Query Type**: **Built-in**
  - **Query**: `SELECT * FROM states`

• Les 50 États des États-Unis sont enregistrés dans la table *states*. Pour que les 50 entrées soient sélectionnées, la requête à effectuer dans la base de données Mysql est la suivante :

  ```sql
  select * from states;
  ```

• Dans l’onglet **Advanced settings**, sélectionnez l’encodage adéquat.

• Puis sélectionnez le composant suivant, le **tFlowMeter**, et paramétrez ses propriétés.
• Cochez la case **Use input connection name as label** pour réutiliser le nom de la connexion d’entrée dans le fichier log de sortie (tFileOutputCSV).

• Dans le champ **Mode**, sélectionnez l’option **Absolute** puisqu’il n’y a pas de flux de référence, et aucun seuil ne doit être défini dans le champ **Threshold** pour cet exemple.

• Puis lancez l’éditeur du tMap pour paramétrer les propriétés de filtrage.

• Pour cet exemple, glissez et déposez les colonnes ID et State de la zone Input (à gauche) du tMap vers la zone Output (à droite). Aucune variable n’est utilisée pour cet exemple.

• Dans la table du flux de sortie (ici, la table *filtered_states*), cliquez sur la flèche dotée d’un (+) pour ajouter un ligne de filtre.

• Glissez la colonne *State* de la zone Input (*row2*) vers le champ filtre et saisissez un filtre permettant de n’obtenir que les États commençant par la lettre *M*. L’expression finale ressemble à : 
  
  `row2.State.startsWith("M")`

• Cliquez sur **OK** pour valider les paramètres.

• Puis sélectionnez le deuxième composant tFlowMeter et paramétrez ses propriétés.
• Cochez la case **Use input connection name as label**.

• Dans le champ **Mode**, sélectionnez l’option **Relative** et dans la liste déroulante **Reference connection**, sélectionnez **US_States** comme référence.

• Une fois de plus, aucun seuil n’est utilisé pour cet exemple, ainsi laissez le champ **Threshold** vide.

• Aucun paramètre particulier n’est requis pour le composant **tLogRow**.

• Il en va de même pour le composant **tFlowMeterCatcher** puisque ses propriétés sont limitées au schéma prédéfini contenant les informations de log.

• Paramétrez les propriétés du composant de sortie (**tFileOutputCSV**).

• Cochez la case **Append** pour logger toutes les mesures du **tFlowMeter**.

• Enregistrez le Job et exécutez-le.

```text
19|Maine
20|Maryland
21|Massachusetts
22|Michigan
23|Minnesota
24|Mississippi
25|Missouri
26|Montana
Job FlowMeterCatcher ended at 17:55 29/08/2007. [exit code=0]
```

La vue **Run** affiche le nom des Etats filtrés tel qu’il a été défini dans le Job.
Dans le fichier csv délimité, le nombre de lignes de la colonne `count` varie entre le composant `tFlowMeter1` et le `tFlowMeter2` puisque les données ont été filtrées. La colonne `reference` affiche cette différence.

<table>
<thead>
<tr>
<th>A</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>moment</td>
<td>origin</td>
<td>label</td>
<td>count</td>
<td>reference</td>
</tr>
<tr>
<td>2</td>
<td>08/29/07 05:56 PM</td>
<td>tFlowMeter_1</td>
<td>US_States</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>08/29/07 05:56 PM</td>
<td>tFlowMeter_2</td>
<td>filtered_states</td>
<td>8</td>
<td>60</td>
</tr>
</tbody>
</table>
tFlowToIterate

Ce composant lit des données ligne par ligne et stocke les entrées des données dans des variables globales itératives.

Propriétés du tFlowToIterate Standard

Ces propriétés sont utilisées pour configurer le tFlowToIterate s'exécutant dans le framework de Jobs Standard.

Le composant tFlowToIterate Standard appartient à la famille Orchestration.

Le composant de ce framework est toujours disponible.

Basic settings

Use the default (key, value) in global variables  Lorsque cette case est cochée, le système utilise la valeur par défaut de la variable globale utilisée dans le Job.

Customize

key : Entrez le nom de la nouvelle variable globale. Appuyez sur Ctrl+Espace pour accéder à l'ensemble des variables disponibles, qu'elles soient globales ou définies par l'utilisateur.

value : Cliquez dans la cellule pour accéder à la liste des colonnes rattachées à la variable globale définie.

Advanced settings

tStatCatcher Statistics  Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.

Global Variables

Global Variables

ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

CURRENT_ITERATION : numéro de séquence de l'itération courante. Cette variable est une variable Flow et retourne un entier.

Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette
 liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Le tFlowToIterate n’est pas un composant de début, et il requiert un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connections</td>
<td>Liens de sortie (de ce composant à un autre) :</td>
</tr>
<tr>
<td></td>
<td><strong>Row</strong> : Iterate.</td>
</tr>
<tr>
<td></td>
<td><strong>Trigger</strong> : Run if, OnComponentOk, OnComponentError.</td>
</tr>
<tr>
<td></td>
<td>Liens d’entrée (d’un autre composant à celui-ci) :</td>
</tr>
<tr>
<td></td>
<td><strong>Row</strong> : Main.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les liens, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Scénario : Transformer un flux de données en liste**

Le scénario suivant décrit un Job permettant de lire une liste de fichiers à partir d’un fichier d’entrée donné. Il effectue également une itération sur chacun des fichiers et affiche leur contenu ligne par ligne dans la console de log Run.

**Construire le Job**

**Procédure**

1. A partir de la Palette, cliquez-déposez les composants suivants dans l’espace de modélisation : deux tFileInputDelimited, un tFlowToIterate et un tLogRow.

2. Cliquez-droit sur chacun des composants pour les relier entre eux. Connectez le premier tFileInputDelimited au tFlowToIterate à l’aide d’un lien de type Row Main, puis le tFlowToIterate au deuxième tFileInputDelimited à l’aide d’un lien de type Iterate, enfin le second tFileInputDelimited au tLogRow à l’aide d’un lien de type Row Main.

**Résultats**
Configurer les composants

Procédure

1. Dans l’espace de modélisation graphique, double-cliquez sur le premier **tFileInputDelimited** pour afficher sa vue **Component**.

2. Dans l’onglet **Basic settings**, cliquez sur le bouton [...] situé à côté du champ **File Name** pour déterminer le chemin d’accès au fichier d’entrée.

   A **Remarque** :
   
   Le champ **File Name** est obligatoire.

3. Cliquez sur le bouton **Edit schema** pour décrire la structure des données du fichier d’entrée. Dans ce scénario, le schéma comporte une seule colonne appelée **FileName**.

4. Dans l’espace de modélisation graphique, sélectionnez le **tFlowToIterate** et cliquez sur l’onglet **Component** pour définir sa configuration de base (**Basic settings**).
Cliquez sur le bouton [+] pour ajouter des lignes de paramètres et définissez vos variables. Cliquez dans la cellule key afin de saisir le nom de la variable souhaité. Dans ce scénario, la variable est la suivante : "Name_of_File".
Vous pouvez également cocher la case Use the default (key, value) in global variables pour utiliser la valeur par défaut des variables globales.

5. Dans l'espace de modélisation graphique, double-cliquez sur le second composant tFileInputDelimited pour ouvrir sa vue Component.

Dans le champ File Name, renseignez le répertoire des fichiers à lire puis appuyez sur les touches Ctrl+Espace pour sélectionner la variable globale "Name_of_File". Dans ce scénario, la syntaxe est la suivante :

"C:/scenario/flow_to_iterate/*+(String)globalMap.get("Name_of_File")")

Cliquez sur Edit schema afin de définir le nom de la colonne du schéma. Nommez-la, dans ce scénario, RowContent.
Renseignez les autres champs selon vos besoins. Pour plus d’informations, consultez Propriétés du tFileInputDelimited Standard à la page 1067.

6. Dans l'espace de modélisation graphique, double-cliquez sur le dernier composant, le tLogRow, pour afficher sa vue Component et définir sa configuration de base (Basic settings).
Définissez les paramètres selon vos besoins. Pour plus d’informations, consultez Propriétés du tLogRow Standard à la page 2105.

Sauvegarder et exécuter le Job

Procédure

1. Enregistrez votre Job en appuyant sur les touches Ctrl+S.
2. Appuyez sur F6 pour l’exécuter ou cliquez sur le bouton Run dans la vue Run.

Résultats

Starting job Practice_8_27 at 16:12 28/08/2012.

[statistics] connecting to socket on port 3580
[statistics] connected
RowContent: Madison Moore
RowContent: Andew Taylor
RowContent: Christopher Anderson
RowContent: madison_moore@hotmail.com
RowContent: andre_taylor@usamail.com
RowContent: christopher_anderson@hotmail.com
RowContent: Madison Moore
RowContent: 100 MAIN ST
RowContent: PO BOX 1022
RowContent: SEATTLE WA 98104
RowContent: USA
RowContent: Andrew Taylor
RowContent: 300 BOVISTON AVE E
RowContent: SEATTLE WA 98102
RowContent: USA
RowContent: Christopher Anderson
RowContent: 500 DOCTOR ST
RowContent: SEATTLE WA 98100
RowContent: USA
[statistics] disconnected
Job Practice_8_27 ended at 16:12 28/08/2012. [exit code=0]

La console affiche, pour chaque ligne, le nom du client, son e-mail et son adresse, précédés du nom de la colonne du schéma.
tForeach

Ce composant crée une boucle sur une liste pour un lien Iterate.

Propriétés du tForeach Standard

Ces propriétés sont utilisées pour configurer le tForeach s'exécutant dans le framework de Jobs Standard.
Le composant tForeach Standard appartient à la famille Orchestration.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Values</th>
<th>Utilisez le bouton [+] pour ajouter des lignes au tableau Values. Cliquez ensuite sur les champs afin de saisir, entre guillemets, les valeurs de la liste sur laquelle effectuer une boucle.</th>
</tr>
</thead>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
CURRENT_VALUE : valeur sur laquelle s’effectue l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères.
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |
|-----------------|---------------------------------------------------------------------------------------------------------------|

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Le tForeach est un composant d’entrée. Un lien Iterate est requis pour relier ce composant à un autre.</th>
</tr>
</thead>
</table>
Scénario : Faire une boucle sur une liste et récupérer les valeurs

Ce scénario décrit un Job à deux composants dans lequel une liste est créée et sur laquelle est effectuée une boucle, via un composant `tForeach`. Les valeurs sont récupérées par un composant `tJava`.

Construire le Job

Procédure

1. Déposez dans l’espace de modélisation graphique les composants `tForeach` et `tJava`.
2. Reliez le `tForeach` au `tJava` à l’aide d’un lien `Row > Iterate`.

Résultats

Configurer les composants

Procédure

1. Double-cliquez sur le `tForeach` afin d’ouvrir sa vue `Basic settings`.
2. Cliquez sur le bouton pour ajouter autant de lignes que nécessaire dans le tableau `Values`.
3. Cliquez sur les champs `Value` afin de saisir les valeurs de la liste, entre guillemets doubles.
4. Double-cliquez sur le `tJava` afin d’ouvrir sa vue `Basic settings`.
5. Saisissez le code Java suivant dans la zone **Code**:
```
System.out.println(globalMap.get("tForeach_1_CURRENT_VALUE") +"_out");
```

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** pour l’exécuter.

**Résultats**

La console affiche la liste des valeurs récupérées du **tForeach**, chaque valeur ayant le suffixe _out :

```
[statistics] connecting to socket on port 3409
[statistics] connected
element_1_out
element_2_out
element_3_out
element_4_out
[statistics] disconnected
Job tForeach_2 ended at 14:28
04/10/2010. [exit code=0]
```
tFTPClose

Ce composant ferme une connexion active à un FTP afin de libérer les ressources occupées.

Propriétés du tFTPClose Standard

Ces propriétés sont utilisées pour configurer le tFTPClose s’exécutant dans le framework de Jobs Standard.
Le composant tFTPClose Standard appartient à la famille Internet.
Le composant de ce framework est toujours disponible.

Basic settings

| Component list | Sélectionnez dans la liste le composant établissant la connexion à fermer. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Global Variables

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants FTP, particulièrement avec le composant tFTPConnection. |

Scénarios associés

- Lister et obtenir des fichiers/dossiers d’un répertoire FTP à la page 1299.
- Placer des fichiers sur un serveur FTP à la page 1316.
- Renommer un fichier situé sur un serveur FTP à la page 1323.
tFTPConnection

Ce composant ouvre une connexion FTP afin d'effectuer des transferts de fichiers en une seule transaction.

Propriétés du tFTPConnection Standard

Ces propriétés sont utilisées pour configurer le tFTPConnection s'exécutant dans le framework de Jobs Standard.

Le composant tFTPConnection Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Type</strong></td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td>Host</td>
<td>Saisissez l’adresse IP ou le nom de l’hôte du serveur FTP.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute du serveur FTP.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les données d’authentification de l’utilisateur au serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>SFTP Support</td>
<td>Cochez cette case pour vous connecter au serveur FTP via une connexion SFTP.</td>
</tr>
<tr>
<td>Authentication method</td>
<td>Sélectionnez la méthode d’authentification SFTP, Public key ou Password.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Public key</strong> : saisissez le chemin d’accès à la clé privée et la phrase secrète pour la clé, dans les champs Private key et Key Passphrase, respectivement.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Password</strong> : saisissez le mot de passe requis.</td>
</tr>
<tr>
<td><strong>tFTPConnection</strong></td>
<td>Cette propriété est disponible uniquement lorsque la case <strong>SFTP Support</strong> est cochée.</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Filename encoding</strong></td>
<td>Cochez cette case pour paramétrer l’encodage utilisé pour convertir les noms de fichiers de chaînes de caractères à octets. L’encodage utilisé doit être le même que celui utilisé sur le serveur SFTP. Si la version du serveur SFTP est supérieure à 3, l’encodage doit être UTF-8. Si ce n’est pas le cas, une erreur survient. Cette propriété est disponible uniquement lorsque la case <strong>SFTP Support</strong> est cochée.</td>
</tr>
<tr>
<td><strong>FTPS Support</strong></td>
<td>Cochez cette case pour vous connecter au serveur FTP via une connexion FTPS.</td>
</tr>
<tr>
<td><strong>Keystore File</strong></td>
<td>Saisissez le chemin d’accès au fichier Keystore (fichier protégé par un mot de passe et contenant plusieurs clés et certificats). Cette propriété est disponible uniquement lorsque la case <strong>FTPS Support</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Keystore Password</strong></td>
<td>Saisissez votre mot de passe Keystore. Cette propriété est disponible uniquement lorsque la case <strong>FTPS Support</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Security Mode</strong></td>
<td>Sélectionnez le mode de sécurité dans la liste <strong>Implicit</strong> or <strong>Explicit</strong>. Cette propriété est disponible uniquement lorsque la case <strong>FTPS Support</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Connection mode</strong></td>
<td>Sélectionnez dans la liste le mode de connexion, <strong>Passive</strong> ou <strong>Active</strong>.</td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
<td>Sélectionnez dans la liste un type d’encodage ou sélectionnez <strong>CUSTOM</strong> et définissez-le manuellement.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>Use Socks Proxy</strong></th>
<th>Cochez cette case si vous utilisez un proxy et, dans les champs <strong>Proxy host</strong>, <strong>Proxy port</strong>, <strong>Proxy user</strong> et <strong>Proxy password</strong> qui s’affichent, spécifiez les paramètres du serveur proxy.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Channel Protection Level</strong></td>
<td>Niveau de protection du canal de données avec lequel les données sont transférées entre le client et le serveur. Pour plus d’informations, consultez RFC 2228: FTP Security Extensions (en anglais). Cette propriété est disponible uniquement lorsque la case <strong>FTPS Support</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Protection Buffer Size</strong></td>
<td>Taille maximale (en octets) des blocs de données encodées à transférer entre le client et le serveur. Pour plus d’informations, consultez RFC 2228: FTP Security Extensions (en anglais).</td>
</tr>
</tbody>
</table>
Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée.

**tStatCatcher Statistics**
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Variables globales**

**ERROR_MESSAGE**
Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

**Utilisation**

**Règle d’utilisation**
Ce composant est généralement utilisé comme sous-job a un seul composant.

**Scénarios associés**

- Lister et obtenir des fichiers/dossiers d’un répertoire FTP à la page 1299.
- Placer des fichiers sur un serveur FTP à la page 1316.
- Renommer un fichier situé sur un serveur FTP à la page 1323.
tFTPDelete

Ce composant supprime des fichiers ou dossiers d’un répertoire spécifié sur un serveur FTP.

Propriétés du tFTPDelete Standard

Ces propriétés sont utilisées pour configurer le tFTPDelete s’exécutant dans le framework de Jobs Standard.

Le composant tFTPDelete Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Built-In : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• Repository : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Saisissez l’adresse IP ou le nom de l’hôte du serveur FTP.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute du serveur FTP.</td>
</tr>
</tbody>
</table>
| Username et Password        | Saisissez les données d’authentification de l’utilisateur au serveur FTP.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
<p>| Remote directory            | Répertoire où se trouvent les fichiers/dossiers à supprimer. |
| Move to the current directory | Cochez cette case pour modifier le répertoire en l’un des répertoires spécifiés dans le champ Remote directory. Le composant FTP suivant dans le Job prendra ce répertoire comme racine du répertoire distant lors de l’utilisation de la même connexion. |</p>
<table>
<thead>
<tr>
<th><strong>tFTPDelete</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cette propriété est disponible uniquement lorsque la case Use an existing connection est cochée.</td>
</tr>
<tr>
<td><strong>SFTP Support</strong></td>
</tr>
<tr>
<td>Cochez cette case pour vous connecter au serveur FTP via une connexion SFTP.</td>
</tr>
<tr>
<td><strong>Authentication method</strong></td>
</tr>
</tbody>
</table>
| Sélectionnez la méthode d’authentification SFTP, Public key ou Password.  
  - Public key : saisissez le chemin d’accès à la clé privée et la phrase secrète pour la clé, dans les champs Private key et Key Passphrase, respectivement.  
  - Password : saisissez le mot de passe requis.  
  Cette propriété est disponible uniquement lorsque la case SFTP Support est cochée. |
| **Filename encoding** |
| Cochez cette case pour paramétrer l’encodage utilisé pour convertir les noms de fichiers de chaînes de caractères à octets. L’encodage utilisé doit être le même que celui utilisé sur le serveur SFTP. Si la version du serveur SFTP est supérieure à 3, l’encodage doit être UTF-8. Si ce n’est pas le cas, une erreur survient.  
  Cette propriété est disponible uniquement lorsque la case SFTP Support est cochée. |
| **FTPS Support** |
| Cochez cette case pour vous connecter au serveur FTP via une connexion FTPS. |
| **Keystore File** |
| Saisissez le chemin d’accès au fichier Keystore (fichier protégé par un mot de passe et contenant plusieurs clés et certificats).  
  Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée. |
| **Keystore Password** |
| Saisissez votre mot de passe Keystore.  
  Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée. |
| **Security Mode** |
| Sélectionnez le mode de sécurité dans la liste Implicit ou Explicit.  
  Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée. |
| **Use Perl5 Regex Expression as Filemask** |
| Cochez cette case pour utiliser les expression régulières Perl5 dans le champ Files comme filtres de fichiers. Cela est utile lorsque le nom du fichier à traiter contient des caractères spéciaux, comme des parenthèses.  
  Pour plus d’informations concernant la syntaxe des expressions régulières Perl5, consultez Perl5 Regular Expression Syntax (en anglais). |
<p>| <strong>Files</strong> |
| Noms des fichiers/dossiers ou chemins d’accès aux fichiers/dossiers à supprimer. Vous pouvez spécifier plusieurs fichiers/dossiers dans une ligne en utilisant |</p>
<table>
<thead>
<tr>
<th><strong>Target Type</strong></th>
<th>Sélectionnez le type de cible à supprimer, un fichier (<em>File</em>) ou un répertoire (<em>Directory</em>).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection mode</strong></td>
<td>Sélectionnez dans la liste le mode de connexion, <em>Passive</em> ou <em>Active</em>.</td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
<td>Sélectionnez dans la liste un type d’encodage ou sélectionnez <em>CUSTOM</em> et définissez-le manuellement.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>Use Socks Proxy</strong></th>
<th>Cochez cette case si vous utilisez un proxy et, dans les champs <em>Proxy host</em>, <em>Proxy port</em>, <em>Proxy user</em> et <em>Proxy password</em> qui s’affichent, spécifiez les paramètres du serveur proxy.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ignore Failure At Quit (FTP)</strong></td>
<td>Cochez cette case pour ignorer les erreurs de fermeture de bibliothèques ou les erreurs de fermeture FTP.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Variables globales**

<table>
<thead>
<tr>
<th><strong>ERROR_MESSAGE</strong></th>
<th>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_FILE</strong></td>
<td>Nombre de fichiers traités. Cette variable est une variable After et retourne un entier.</td>
</tr>
</tbody>
</table>


**CURRENT_STATUS**

Résultat d'exécution du composant. Cette variable est une variable Flow et retourne une chaîne de caractères.

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin. |

### Scénario associé

Aucun scénario n’est disponible pour ce composant.
**tFTPFileExist**

Ce composant vérifie la présence d’un fichier ou d’un répertoire sur le serveur FTP.

**Propriétés du tFTPFileExist Standard**

Ces propriétés sont utilisées pour configurer le tFTPFileExist s’exécutant dans le framework de Jobs Standard.

Le composant tFTPFileExist Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Select the way the connection details will be set.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong>: The connection details will be set locally for this component. You need to specify the values for all related connection properties manually.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong>: The connection details stored centrally in Repository &gt; Metadata will be reused by this component. You need to click the [...] button next to it and in the pop-up Repository Content dialog box, select the connection details to be reused, and all related connection properties will be automatically filled in.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Select this check box and in the Component List click the relevant connection component to reuse the connection details you already defined.</td>
</tr>
<tr>
<td>Host</td>
<td>The IP address or hostname of the FTP server.</td>
</tr>
<tr>
<td>Port</td>
<td>The listening port number of the FTP server.</td>
</tr>
<tr>
<td>Username and Password</td>
<td>The user authentication data to access the FTP server. To enter the password, click the [...] button next to the password field, and then in the pop-up dialog box enter the password between double quotes and click OK to save the settings.</td>
</tr>
<tr>
<td>Remote directory</td>
<td>Répertoire distant dans lequel vérifier que le fichier ou le répertoire existe.</td>
</tr>
<tr>
<td>Target Type</td>
<td>Sélectionnez le type de cible à vérifier, File (fichier) ou Directory (répertoire).</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom ou chemin du fichier dont vous voulez vérifier la présence. Cette propriété est disponible uniquement lorsque l’option File est sélectionnée dans la liste Target Type.</td>
</tr>
</tbody>
</table>
| Directory Name          | Nom ou chemin du répertoire dont vous voulez vérifier la présence. 
Cette propriété est disponible uniquement lorsque l’option Directory est sélectionnée dans la liste Target Type. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SFTP Support</td>
<td>Cochez cette case pour vous connecter au serveur FTP via une connexion SFTP.</td>
</tr>
</tbody>
</table>
| Authentication method  | Sélectionnez la méthode d’authentification SFTP, Public key ou Password.  
- Public key : saisissez le chemin d’accès à la clé privée et la phrase secrète pour la clé, dans les champs Private key et Key Passphrase, respectivement.  
- Password : saisissez le mot de passe requis.  
Cette propriété est disponible uniquement lorsque la case SFTP Support est cochée. |
| Filename encoding      | Cochez cette case pour paramétrer l’encodage utilisé pour convertir les noms de fichiers de chaînes de caractères à octets. L’encodage utilisé doit être le même que celui utilisé sur le serveur SFTP. Si la version du serveur SFTP est supérieure à 3, l’encodage doit être UTF-8. Si ce n’est pas le cas, une erreur survient.  
Cette propriété est disponible uniquement lorsque la case SFTP Support est cochée. |
| FTPS Support           | Cochez cette case pour vous connecter au serveur FTP via une connexion FTPS.                                                                                                                    |
| Keystore File          | Saisissez le chemin d’accès au fichier Keystore (fichier protégé par un mot de passe et contenant plusieurs clés et certificats).  
Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée. |
| Keystore Password      | Saisissez votre mot de passe Keystore.  
Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée.                                                                                                           |
| Security Mode          | Sélectionnez le mode de sécurité dans la liste Implicit ou Explicit.  
Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée.                                                                                                           |
| Connection mode        | Sélectionnez dans la liste le mode de connexion, Passive ou Active.                                                                                                                              |
| Encoding               | Sélectionnez dans la liste un type d’encodage ou sélectionnez CUSTOM et définissez-le manuellement.                                                                                               |
**Advanced settings**

<table>
<thead>
<tr>
<th>Use Socks Proxy</th>
<th>Cochez cette case si vous utilisez un proxy et, dans les champs Proxy host, Proxy port, Proxy user et Proxy password qui s’affichent, spécifiez les paramètres du serveur proxy.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore Failure At Quit (FTP)</td>
<td>Cochez cette case pour ignorer les erreurs de fermeture de bibliothèques ou les erreurs de fermeture FTP.</td>
</tr>
<tr>
<td>Data Channel Protection Level</td>
<td>Niveau de protection du canal de données avec lequel les données sont transférées entre le client et le serveur. Pour plus d’informations, consultez RFC 2228: FTP Security Extensions (en anglais). Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Variables globales**

<table>
<thead>
<tr>
<th>ERROR_MESSAGE</th>
<th>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXISTS</td>
<td>Indique si un fichier/dossier spécifié existe. Cette variable est une variable Flow et retourne un booléen.</td>
</tr>
<tr>
<td>FILENAME</td>
<td>Nom du fichier/répertoire traité. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin.</th>
</tr>
</thead>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tFTPFileList

Ce composant liste tous les fichiers et dossiers directement dans un répertoire spécifié, en se basant sur un modèle de masque de fichier.

Propriétés du tFTPFileList Standard

Ces propriétés sont utilisées pour configurer le tFTPFileList s'exécutant dans le framework de Jobs Standard.
Le composant tFTPFileList Standard appartient à la famille Internet.
Le composant de ce framework est toujours disponible.

Basic settings

| Property Type | Sélectionnez la manière de configurer les informations de connexion.
|              | • Built-In : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.
|              | • Repository : les informations de connexion stockées centralement dans le Repository > Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.

Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.

Host | Saisissez l'adresse IP ou le nom de l'hôte du serveur FTP.

Port | Saisissez le numéro du port d’écoute du serveur FTP.

Username et Password | Saisissez les données d'authentification de l'utilisateur au serveur FTP.
| Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

Remote directory | Répertoire distant où se trouvent les fichiers et dossiers à lister.

Move to the current directory | Cochez cette case pour modifier le répertoire en l'un des répertoires spécifiés dans le champ Remote directory. Le composant FTP suivant dans le Job prendra ce répertoire
comme racine du répertoire distant lors de l'utilisation de la même connexion. Cette propriété est disponible uniquement lorsque la case *Use an existing connection* est cochée.

| File detail | Cochez cette case pour lister les détails de chaque fichier/dossier. Ces informations contiennent notamment les droits relatifs au fichier/dossier, le nom de la personne l’ayant créé, le nom du groupe d’utilisateurs ayant des autorisations en lecture/écriture, la taille du fichier et la date de dernière modification. |
| Files | Noms des fichiers/dossiers à lister. Vous pouvez spécifier plusieurs fichiers/dossier sur une ligne en utilisant des caractères de remplacement ou une expression régulière. |
| SFTP Support | Cochez cette case pour vous connecter au serveur FTP via une connexion SFTP. |
| Authentication method | Sélectionnez la méthode d’authentification SFTP, *Public key* ou *Password*.  
• *Public key* : saisissez le chemin d’accès à la clé privée et la phrase secrète pour la clé, dans les champs *Private key* et *Key Passphrase*, respectivement.  
• *Password* : saisissez le mot de passe requis. Cette propriété est disponible uniquement lorsque la case *SFTP Support* est cochée. |
| Filename encoding | Cochez cette case pour paramétrer l’encodage utilisé pour convertir les noms de fichiers de chaînes de caractères à octets. L’encodage utilisé doit être le même que celui utilisé sur le serveur SFTP. Si la version du serveur SFTP est supérieure à 3, l’encodage doit être *UTF-8*. Si ce n’est pas le cas, une erreur survient. Cette propriété est disponible uniquement lorsque la case *SFTP Support* est cochée. |
| FTPS Support | Cochez cette case pour vous connecter au serveur FTP via une connexion FTPS. |
| Keystore File | Saisissez le chemin d’accès au fichier Keystore (fichier protégé par un mot de passe contenant plusieurs clés et certificats). Cette propriété est disponible uniquement lorsque la case *FTPS Support* est cochée. |
| Keystore Password | Saisissez votre mot de passe Keystore. Cette propriété est disponible uniquement lorsque la case *FTPS Support* est cochée. |
| Security Mode | Sélectionnez le mode de sécurité dans la liste *Implicit* ou *Explicit*. |
**tFTPFileList**

Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée.

### Connection mode

Sélectionnez dans la liste le mode de connexion, **Passive** ou **Active**.

### Encoding

Sélectionnez dans la liste un type d’encodage ou sélectionnez **CUSTOM** et définissez-le manuellement.

### Advanced settings

#### Use Socks Proxy

Cochez cette case si vous utilisez un proxy et, dans les champs **Proxy host**, **Proxy port**, **Proxy user** et **Proxy password** qui s’affichent, spécifiez les paramètres du serveur proxy.

#### Ignore Failure At Quit (FTP)

Cochez cette case pour ignorer les erreurs de fermeture de bibliothèques ou les erreurs de fermeture FTP.

#### Data Channel Protection Level

Niveau de protection du canal de données avec lequel les données sont transférées entre le client et le serveur. Pour plus d’informations, consultez [RFC 2228: FTP Security Extensions](en anglais). Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée.

#### Protection Buffer Size

Taille maximale (en octets) des blocs de données encodées à transférer entre le client et le serveur. Pour plus d’informations, consultez [RFC 2228: FTP Security Extensions](en anglais). Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée.

#### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

#### ERROR_MESSAGE

Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

#### CURRENT_FILE

Nom du fichier courant. Cette variable est une variable Flow et retourne une chaîne de caractères.

#### CURRENT_FILEPATH

Chemin du fichier courant. Cette variable est une variable Flow et retourne une chaîne de caractères.

#### NB_FILE

Nombre de fichiers traités. Cette variable est une variable After et retourne un entier.
Utilisation

| Règle d'utilisation | Le composant tFTPFilelist est généralement utilisé comme composant seul dans un sous-job mais il peut aussi être utilisé avec d'autres composants. |

Lister et obtenir des fichiers/dossiers d'un répertoire FTP

Voici un exemple d'utilisation des composants FTP de Talend pour effectuer des itérations, lister tous les fichiers et dossiers d'un répertoire sur un serveur FTP et récupérer uniquement les fichiers texte de ce répertoire vers un répertoire local.

Créer un Job pour lister et obtenir des fichiers/dossiers d'un répertoire FTP

Créez un Job pour vous connecter à un serveur FTP, effectuer des itérations et récupérer tous les fichiers et dossiers d'un répertoire FTP racine vers un répertoire local, puis fermer la connexion au serveur.

Avant de commencer

Prérequis : pour reproduire ce scénario, un serveur FTP doit être démarré et quelques fichiers/dossiers doivent se situer dans le répertoire racine de ce serveur FTP.
Procédure

1. Créez un nouveau Job et ajoutez un composant **tFTPConnection**, un **tFTPFileList**, un **tIterateToFlow** , un **tLogRow**, un **tFTPGet** et un **tFTPClose**, en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.

2. Reliez le **tFTPFileList** au **tIterateToFlow**, à l’aide d’un lien **Row > Iterate**.

3. Reliez le **tIterateToFlow** au **tLogRow** à l’aide d’un lien **Row > Main**.

4. Connectez le **tFTPConnection** au **tFTPFileList** à l’aide d’un lien **Trigger > OnSubjobOk**.

5. Répétez l’opération pour relier le **tFTPFileList** au **tFTPGet** et le **tFTPGet** au **tFTPClose**.

**Ouvrir une connexion au serveur FTP**

Configurez le composant **tFTPConnection** pour ouvrir une connexion au serveur FTP.

Procédure

1. Double-cliquez sur le **tFTPConnection** pour ouvrir sa vue **Basic settings**.

2. Dans les champs **Host** et **Port**, saisissez l’adresse IP du serveur FTP et le numéro du port d’écoute, respectivement.

3. Dans les champs **Username** et **Password**, saisissez les informations d’authentification.

**Lister tous les fichiers/dossiers du répertoire racine FTP**

Configurez le composant **tFTPFileList**, le **tIterateToFlow** et le **tLogRow** pour effectuer des itérations sur tous les fichiers et dossiers du répertoire racine du FTP, ainsi que pour afficher les noms de et les chemins d’accès à ces fichiers et dossiers dans la console du Studio Talend.

Procédure

1. Double-cliquez sur le composant **tFTPFileList** pour ouvrir sa vue **Basic settings**.
2. Spécifiez les informations de connexion requises pour accéder au serveur FTP. Dans cet exemple, cochez la case **Use an existing connection** et, dans la liste **Component list** qui s’affiche, sélectionnez le composant de connexion afin de réutiliser les informations de connexion précédemment définies.

3. Dans le champ **Remote directory**, spécifiez le répertoire du serveur FTP sur lequel les fichiers et dossier seront itérés. Dans cet exemple, spécifiez `/`, ce qui signifie le répertoire racine du serveur FTP.

4. Décrochez la case **Move to the current directory**.

5. Double-cliquez sur le composant **tIterateToFlow** pour ouvrir sa vue **Basic settings**.

6. Cliquez sur le bouton à côté du champ **Edit schema** pour ouvrir le schéma.
7. Cliquez deux fois sur le bouton pour ajouter deux colonnes de type String, respectivement nommées filename et filepath, qui contiendront respectivement les noms et chemins des fichiers à itérer. Cela fait, cliquez sur OK pour fermer le schéma.

8. Dans la table Mapping, configurez les valeurs pour les colonnes filename et filepath. Dans cet exemple, la variable globale ((String)globalMap.get("tFTPFileList_1_CURRENT_FILE")) pour filename et la variable globale ((String)globalMap.get("tFTPFileList_1_CURRENT_FILEPATH")) pour filepath.

Notez que vous pouvez renseigner les valeurs en appuyant sur les touches Ctrl + Espace, pour accéder à la liste des variables globales. Sélectionnez tFTPFileList_1_CURRENT_FILE et tFTPFileList_1_CURRENT_FILEPATH dans la liste.

9. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings, puis sélectionnez Table (print values in cells of a table) dans la zone Mode, pour une meilleure lisibilité des résultats.

Obtenir des fichiers du répertoire du serveur FTP vers un répertoire local

Configurez le composant tFTPGet pour récupérer uniquement les fichiers texte du répertoire racine du FTP vers un répertoire local.

Procédure

1. Double-cliquez sur le tFTPGet pour ouvrir sa vue Basic settings.

2. Spécifiez les informations de connexion requises pour accéder au serveur FTP. Dans cet exemple, cochez la case Use an existing connection et, dans la liste Component list qui s'affiche,
sélectionnez le composant de connexion afin de réutiliser les informations de connexion précédemment définies.

3. Dans le champ **Local directory**, spécifiez le répertoire local dans lequel télécharger les fichiers et dossiers. Dans cet exemple, spécifiez *D:/FtpDownloads*.

4. Dans le champ **Remote directory**, spécifiez le répertoire du serveur FTP sous lequel les fichiers et dossiers seront téléchargés. Dans cet exemple, spécifiez /*, ce qui indique le répertoire racine du serveur FTP.

5. Dans la table **Files**, cliquez sur le bouton [*] pour ajouter une ligne et, dans la colonne **Filemask**, saisissez "*.txt", entre guillemets doubles, pour récupérer uniquement les fichiers texte du répertoire FTP vers le répertoire local.

**Fermer la connexion au serveur FTP**

Configurez le **tFTPClose** pour fermer la connexion au serveur FTP.

**Procédure**

1. Double-cliquez sur le **tFTPClose** pour ouvrir sa vue **Basic settings**.

2. Dans la liste **Component list**, sélectionnez le composant **tFTPConnection** ouvrant la connexion que vous devez fermer. Dans cet exemple, seul un **tFTPConnection** est utilisé, il est donc sélectionné par défaut.

**Exécuter le Job pour lister et obtenir les fichiers/dossiers du répertoire FTP**

Après avoir configuré le Job et ses composants pour lister et obtenir les fichiers/dossiers du répertoire FTP, vous pouvez exécuter le Job et vérifier ses résultats d’exécution.

**Procédure**

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour l’exécuter.
Comme affiché ci-dessus, les noms et chemins d'accès aux fichiers du répertoire racine du serveur FTP sont affichés dans la console. Seuls les fichiers texte sont téléchargés dans le répertoire local spécifié.
tFTPFileProperties

Ce composant récupère les propriétés d’un fichier spécifié sur un serveur FTP.

Propriétés du tFTPFileProperties Standard

Ces propriétés sont utilisées pour configurer le tFTPFileProperties s'exécutant dans le framework de Jobs Standard.

Le composant tFTPFileProperties Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

Schema et Edit schema

Un schéma est une description de lignes et définit les champs à traiter et à passer au composant suivant.

Le schéma de ce composant est en lecture seule. Il décrit les propriétés principales du fichier spécifié. Vous pouvez cliquer sur le bouton [...] à côté du champ Edit schema pour voir le schéma prédéfini contenant les champs suivants :

• **abs_path** : chemin absolu du fichier.
• **dirname** : répertoire du fichier.
• **basename** : nom du fichier.
• **size** : taille du fichier en octets.
• **mtime** : horodatage indiquant la dernière modification du fichier, en millisecondes depuis l'Epoch Unix (00:00:00 UTC, 1er janvier 1970).
• **mtime_string** : date et heure de dernière modification du fichier.

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.
<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Saisissez l’adresse IP ou le nom de l’hôte du serveur FTP.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Saisissez le numéro du port d’écoute du serveur FTP.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez les données d’authentification de l’utilisateur au serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Remote directory</strong></td>
<td>Chemin d’accès au répertoire dans lequel le fichier est disponible.</td>
</tr>
<tr>
<td><strong>File</strong></td>
<td>Nom ou chemin du fichier duquel récupérer les propriétés.</td>
</tr>
<tr>
<td><strong>Transfer mode</strong></td>
<td>Sélectionnez dans la liste le mode de transfert ascii ou binary.</td>
</tr>
<tr>
<td><strong>SFTP Support</strong></td>
<td>Cochez cette case pour vous connecter au serveur FTP via une connexion SFTP.</td>
</tr>
</tbody>
</table>
| **Authentication method** | Sélectionnez la méthode d’authentification SFTP, Public key ou Password.  
• Public key : saisissez le chemin d’accès à la clé privée et la phrase secrète pour la clé, dans les champs Private key et Key Passphrase, respectivement.  
• Password : saisissez le mot de passe requis.Cette propriété est disponible uniquement lorsque la case SFTP Support est cochée. |
| **Filename encoding** | Cochez cette case pour paramétrer l’encodage utilisé pour convertir les noms de fichiers de chaînes de caractères à octets. L’encodage utilisé doit être le même que celui utilisé sur le serveur SFTP. Si la version du serveur SFTP est supérieure à 3, l’encodage doit être UTF-8. Si ce n’est pas le cas, une erreur survient. Cette propriété est disponible uniquement lorsque la case SFTP Support est cochée. |
| **FTPS Support** | Cochez cette case pour vous connecter au serveur FTP via une connexion FTPS. |
| **Keystore File** | Saisissez le chemin d’accès au fichier Keystore (fichier protégé par un mot de passe et contenant plusieurs clés et certificats). Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée. |
| **Keystore Password** | Saisissez votre mot de passe Keystore. Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée. |
### tFTPFileProperties

<table>
<thead>
<tr>
<th>Security Mode</th>
<th>Sélectionnez le mode de sécurité dans la liste <strong>Implicit</strong> or <strong>Explicit</strong>. Cette propriété est disponible uniquement lorsque la case <strong>FTPS Support</strong> est cochée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection mode</td>
<td>Sélectionnez dans la liste le mode de connexion, <strong>Passive</strong> ou <strong>Active</strong>.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Sélectionnez dans la liste un type d’encodage ou sélectionnez <strong>CUSTOM</strong> et définissez-le manuellement.</td>
</tr>
<tr>
<td>Calculate MD5 Hash</td>
<td>Cochez cette case pour vérifier le MD5 du fichier.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>Use Socks Proxy</th>
<th>Cochez cette case si vous utilisez un proxy et, dans les champs <strong>Proxy host</strong>, <strong>Proxy port</strong>, <strong>Proxy user</strong> et <strong>Proxy password</strong> qui s’affichent, spécifiez les paramètres du serveur proxy.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore Failure At Quit (FTP)</td>
<td>Cochez cette case pour ignorer les erreurs de fermeture de bibliothèques ou les erreurs de fermeture FTP.</td>
</tr>
<tr>
<td>Data Channel Protection Level</td>
<td>Niveau de protection du canal de données avec lequel les données sont transférées entre le client et le serveur. Pour plus d’informations, consultez <a href="http://rfc.net/rfc2228">RFC 2228: FTP Security Extensions</a> (en anglais). Cette propriété est disponible uniquement lorsque la case <strong>FTPS Support</strong> est cochée.</td>
</tr>
<tr>
<td>Protection Buffer Size</td>
<td>Taille maximale (en octets) des blocs de données encodées à transférer entre le client et le serveur. Pour plus d’informations, consultez <a href="http://rfc.net/rfc2228">RFC 2228: FTP Security Extensions</a> (en anglais). Cette propriété est disponible uniquement lorsque la case <strong>FTPS Support</strong> est cochée.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Variables globales

<table>
<thead>
<tr>
<th>ERROR_MESSAGE</th>
<th>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
</table>

### Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en standalone.                                                                                                                  |
Scénario associé

Scénario : Afficher les propriétés d’un fichier traité à la page 1221
tFTPGet

Télécharge des fichiers dans un répertoire local depuis un répertoire FTP.

Propriétés du tFTPGet Standard

Ces propriétés sont utilisées pour configurer le tFTPGet s'exécutant dans le framework de Jobs Standard.

Le composant tFTPGet Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |
| Host                       | Saisissez l’adresse IP ou le nom de l’hôte du serveur FTP. |
| Port                       | Saisissez le numéro du port d’écoute du serveur FTP. |
| Username et Password       | Saisissez les données d’authentification de l’utilisateur au serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Local directory            | Répertoire local dans lequel sauvegarder les fichiers téléchargés. |
| Remote directory           | Répertoire FTP duquel télécharger les fichiers. |
| Move to the current directory | Cochez cette case pour modifier le répertoire en l’un des répertoires spécifiés dans le champ Remote directory. Le composant FTP suivant dans le Job prendra ce répertoire... |
comme racine du répertoire distant lors de l'utilisation
de la même connexion.
Cette propriété est disponible uniquement lorsque la
case **Use an existing connection** est cochée.

<table>
<thead>
<tr>
<th><strong>Transfer mode</strong></th>
<th>Sélectionnez dans la liste le mode de transfert <em>ascii</em> ou <em>binary</em>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overwrite file</strong></td>
<td>Sélectionnez l’action à effectuer lorsque le fichier existe déjà.</td>
</tr>
<tr>
<td></td>
<td>• <em>never</em>: ne jamais écraser le fichier.</td>
</tr>
<tr>
<td></td>
<td>• <em>always</em>: toujours écraser le fichier.</td>
</tr>
<tr>
<td></td>
<td>• <em>size different</em>: écraser le fichier lorsque la taille du</td>
</tr>
<tr>
<td></td>
<td>fichier est différente.</td>
</tr>
<tr>
<td></td>
<td>• <em>overwrite</em>: écraser le fichier existant.</td>
</tr>
<tr>
<td></td>
<td>• <em>resume</em>: reprendre le téléchargement du fichier à</td>
</tr>
<tr>
<td></td>
<td>partir du point d’interruption.</td>
</tr>
<tr>
<td></td>
<td>• <em>append</em>: ajouter des données à la fin du fichier sans</td>
</tr>
<tr>
<td></td>
<td>écraser les données.</td>
</tr>
<tr>
<td>Les options <em>overwrite</em>, <em>resume</em> et <em>append</em> sont</td>
<td></td>
</tr>
<tr>
<td>disponibles lorsque la case <strong>SFTP Support</strong> est cochée.</td>
<td></td>
</tr>
<tr>
<td><strong>Append</strong></td>
<td>Cochez cette case pour écrire les données à la fin du</td>
</tr>
<tr>
<td></td>
<td>fichier afin d’éviter l’écrasement des données.</td>
</tr>
<tr>
<td><strong>SFTP Support</strong></td>
<td>Cochez cette case pour vous connecter au serveur FTP</td>
</tr>
<tr>
<td></td>
<td>via une connexion SFTP.</td>
</tr>
<tr>
<td><strong>Authentication method</strong></td>
<td>Sélectionnez la méthode d’authentification SFTP, <em>Public key</em> ou<em>Password</em>.</td>
</tr>
<tr>
<td></td>
<td>• <em>Public key</em>: saisissez le chemin d’accès à la clé privée</td>
</tr>
<tr>
<td></td>
<td>et la phrase secrète pour la clé, dans les champs <em>Private key</em> et <em>Key Passphrase</em>, respectivement.</td>
</tr>
<tr>
<td></td>
<td>• <em>Password</em>: saisissez le mot de passe requis.</td>
</tr>
<tr>
<td>Cette propriété est disponible uniquement lorsque la</td>
<td></td>
</tr>
<tr>
<td>case <strong>SFTP Support</strong> est cochée.</td>
<td></td>
</tr>
<tr>
<td><strong>Filename encoding</strong></td>
<td>Cochez cette case pour paramétrer l’encodage utilisé</td>
</tr>
<tr>
<td></td>
<td>pour convertir les noms de fichiers de chaînes de</td>
</tr>
<tr>
<td></td>
<td>caractères à octets. L’encodage utilisé doit être le même</td>
</tr>
<tr>
<td></td>
<td>que celui utilisé sur le serveur SFTP. Si la version du</td>
</tr>
<tr>
<td></td>
<td>serveur SFTP est supérieure à 3, l’encodage doit être</td>
</tr>
<tr>
<td></td>
<td>UTF-8. Si ce n’est pas le cas, une erreur survient.</td>
</tr>
<tr>
<td>Cette propriété est disponible uniquement lorsque la</td>
<td></td>
</tr>
<tr>
<td>case <strong>SFTP Support</strong> est cochée.</td>
<td></td>
</tr>
<tr>
<td><strong>FTPS Support</strong></td>
<td>Cochez cette case pour vous connecter au serveur FTP</td>
</tr>
<tr>
<td></td>
<td>via une connexion FTPS.</td>
</tr>
<tr>
<td><strong>Keystore File</strong></td>
<td>Saisissez le chemin d’accès au fichier Keystore (fichier</td>
</tr>
<tr>
<td></td>
<td>protégé par un mot de passe et contenant plusieurs clés et certificats).</td>
</tr>
<tr>
<td><strong>tFTPGet</strong></td>
<td>Cette propriété est disponible uniquement lorsque la case <strong>FTPS Support</strong> est cochée.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Keystore Password</strong></td>
<td>Saisissez votre mot de passe Keystore. Cette propriété est disponible uniquement lorsque la case <strong>FTPS Support</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Security Mode</strong></td>
<td>Sélectionnez le mode de sécurité dans la liste <strong>Implicit</strong> ou <strong>Explicit</strong>. Cette propriété est disponible uniquement lorsque la case <strong>FTPS Support</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Use Perl5 Regex Expression as Filemask</strong></td>
<td>Cochez cette case pour utiliser les expression régulières Perl5 dans le champ <strong>Files</strong> comme filtres de fichiers. Cela est utile lorsque le nom du fichier à traiter contient des caractères spéciaux, comme des parenthèses. Pour plus d’informations concernant la syntaxe des expressions régulières Perl5, consultez [Perl5 Regular Expression Syntax](en anglais).</td>
</tr>
<tr>
<td><strong>Files</strong></td>
<td>Noms ou chemins des fichiers à télécharger. Vous pouvez spécifier plusieurs fichiers sur une ligne en utilisant des caractères de remplacement ou une expression régulière.</td>
</tr>
<tr>
<td><strong>Connection mode</strong></td>
<td>Sélectionnez dans la liste le mode de connexion, <strong>Passive</strong> ou <strong>Active</strong>.</td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
<td>Sélectionnez dans la liste un type d’encodage ou sélectionnez <strong>CUSTOM</strong> et définissez-le manuellement.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus.</td>
</tr>
</tbody>
</table>

### Advanced settings

| **Use Socks Proxy** | Cochez cette case si vous utilisez un proxy et, dans les champs **Proxy host**, **Proxy port**, **Proxy user** et **Proxy password** qui s’affichent, spécifiez les paramètres du serveur proxy. |
| **Ignore Failure At Quit (FTP)** | Cochez cette case pour ignorer les erreurs de fermeture de bibliothèques ou les erreurs de fermeture FTP. |
| **Print message** | Cochez cette case pour afficher la liste des fichiers téléchargés dans la console. |
| **Data Channel Protection Level** | Niveau de protection du canal de données avec lequel les données sont transférées entre le client et le serveur. Pour plus d’informations, consultez [RFC 2228: FTP Security Extensions](en anglais). Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée. |
**Protection Buffer Size**
Taille maximale (en octets) des blocs de données encodées à transférer entre le client et le serveur. Pour plus d'informations, consultez [RFC 2228: FTP Security Extensions](en anglais). Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée.

**tStatCatcher Statistics**
Cochez cette case pour collecter les métdonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.

### Variables globales

| **ERROR_MESSAGE** | Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |
| **NB_FILE** | Nombre de fichiers traités. Cette variable est une variable After et retourne un entier. |
| **CURRENT_STATUS** | Résultat d'exécution du composant. Cette variable est une variable Flow et retourne une chaîne de caractères. |
| **TRANSFER_MESSAGES** | Informations transférées du fichier. Cette variable est une variable After et retourne une chaîne de caractères. |

### Utilisation

**Règle d'utilisation**
Ce composant est généralement utilisé comme sous-Job a un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin.

### Scénario associé

Lister et obtenir des fichiers/dossiers d'un répertoire FTP à la page 1299
tFTPPut

Charge des fichiers d’un répertoire local vers un répertoire FTP.

Propriétés du tFTPPut Standard

Ces propriétés sont utilisées pour configurer le tFTPPut s’exécutant dans le framework de Jobs Standard.

Le composant tFTPPut Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Saisissez l’adresse IP ou le nom de l’hôte du serveur FTP.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute du serveur FTP.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les données d’authentification de l’utilisateur au serveur FTP. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Local directory</td>
<td>The local directory from which the files will be uploaded to the FTP server.</td>
</tr>
<tr>
<td>Remote directory</td>
<td>The FTP directory where the uploaded files will be placed.</td>
</tr>
<tr>
<td>Move to the current directory</td>
<td>Cochez cette case pour modifier le répertoire en l’un des répertoires spécifiés dans le champ <strong>Remote directory</strong>.</td>
</tr>
<tr>
<td>composant FTP suivant dans le Job prendra ce répertoire comme racine du répertoire distant lors de l'utilisation de la même connexion. Cette propriété est disponible uniquement lorsque la case Use an existing connection est cochée.</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>Transfer mode</strong></td>
<td>Sélectionnez dans la liste le mode de transfert ascii ou binary.</td>
</tr>
</tbody>
</table>
| **Overwrite file** | Sélectionnez l’action à effectuer lorsque le fichier existe déjà.  
- **never**: ne jamais écraser le fichier.  
- **always**: toujours écraser le fichier.  
- **size different**: écraser le fichier lorsque la taille du fichier est différente.  
- **overwrite**: écraser le fichier existant.  
- **resume**: reprendre le téléchargement du fichier à partir du point d’interruption.  
- **append**: ajouter des données à la fin du fichier sans écraser les données.  
Les options overwrite, resume et append sont disponibles lorsque la case SFTP Support est cochée. |
| **Append** | Cochez cette case pour écrire les données à la fin du fichier afin d’éviter l’écrasement des données. |
| **SFTP Support** | Cochez cette case pour vous connecter au serveur FTP via une connexion SFTP. |
| **Authentication method** | Sélectionnez la méthode d’authentification SFTP, Public key ou Password.  
- **Public key**: saisissez le chemin d’accès à la clé privée et la phrase secrète pour la clé, dans les champs Private key et Key Passphrase, respectivement.  
- **Password**: saisissez le mot de passe requis.  
Cette propriété est disponible uniquement lorsque la case SFTP Support est cochée. |
| **Filename encoding** | Cochez cette case pour paramétrer l’encodage utilisé pour convertir les noms de fichiers de chaînes de caractères à octets. L’encodage utilisé doit être le même que celui utilisé sur le serveur SFTP. Si la version du serveur SFTP est supérieure à 3, l’encodage doit être UTF-8. Si ce n’est pas le cas, une erreur survient.  
Cette propriété est disponible uniquement lorsque la case SFTP Support est cochée. |
| **FTPS Support** | Cochez cette case pour vous connecter au serveur FTP via une connexion FTPS. |
| **Keystore File** | Saisissez le chemin d’accès au fichier Keystore (fichier protégé par un mot de passe et contenant plusieurs clés et certificats). |
Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée.

### Keystore Password
Saisissez votre mot de passe Keystore.
Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée.

### Security Mode
 Sélectionnez le mode de sécurité dans la liste **Implicit** or **Explicit**.
Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée.

### Use Perl5 Regex Expression as Filemask
Cochez cette case pour utiliser les expression régulières Perl5 dans le champ **Files** comme filtres de fichiers. Cela est utile lorsque le nom du fichier à traiter contient des caractères spéciaux, comme des parenthèses.
Pour plus d’informations concernant la syntaxe des expressions régulières Perl5, consultez [Perl5 Regular Expression Syntax](en anglais).

### Use Perl5 Regex Expression as Filemask
Cochez cette case pour utiliser les expression régulières Perl5 dans le champ **Files** comme filtres de fichiers. Cela est utile lorsque le nom du fichier à traiter contient des caractères spéciaux, comme des parenthèses.
Pour plus d’informations concernant la syntaxe des expressions régulières Perl5, consultez [Perl5 Regular Expression Syntax](en anglais).

### Files
Spécifiez les fichiers à télécharger.
- **Filemask**: les noms ou chemins de fichiers à télécharger.
- **New name**: nom à donner au fichier après le transfert.

### Connection mode
 Sélectionnez dans la liste le mode de connexion, **Passive** ou **Active**.

### Encoding
 Sélectionnez dans la liste un type d’encodage ou sélectionnez **CUSTOM** et définissez-le manuellement.

### Die on error
Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.
Décochez la case pour ignorer les lignes en erreur et terminer le processus.

### Advanced settings

#### Use Socks Proxy
Cochez cette case si vous utilisez un proxy et, dans les champs **Proxy host**, **Proxy port**, **Proxy user** et **Proxy password** qui s’affichent, spécifiez les paramètres du serveur proxy.

#### Ignore Failure At Quit (FTP)
Cochez cette case pour ignorer les erreurs de fermeture de bibliothèques ou les erreurs de fermeture FTP.


tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

Variables globales

<table>
<thead>
<tr>
<th>ERROR_MESSAGE</th>
<th>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_FILE</td>
<td>Nombre de fichiers traités. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>CURRENT_STATUS</td>
<td>Résultat d’exécution du composant. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>CURRENT_FILE_EXISTS</td>
<td>Indique si le fichier courant existe. Cette variable est une variable Flow et retourne un booléen.</td>
</tr>
<tr>
<td>TRANSFER_MESSAGES</td>
<td>Informations transférées du fichier. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme sous-Job a un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin. |

Placer des fichiers sur un serveur FTP

Voici un exemple d’utilisation des composants FTP de Talend pour placer des fichiers dans un répertoire local sur un serveur FTP.

Créer un Job pour placer des fichiers sur un serveur FTP

Créez un Job pour vous connecter à un serveur FTP, placer différents fichiers locaux sur le serveur puis fermer la connexion au serveur.
**Procédure**

1. Créez un nouveau Job et ajoutez un composant `tFTPConnection`, un `tFTPPut` et un `tFTPClose` en saisisant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.

2. Reliez le `tFTPConnection` au `tFTPPut` à l’aide d’un lien Trigger > OnSubjobOk.

3. Reliez le `tFTPPut` au `tFTPClose` à l’aide d’un lien Trigger > OnSubjobOk.

**Ouvrir une connexion au serveur FTP**

Configurez le `tFTPConnection` pour ouvrir une connexion au serveur FTP.

**Procédure**

1. Double-cliquez sur le `tFTPConnection` pour ouvrir sa vue Basic settings.

2. Dans les champs **Host** et **Port**, saisissez l’adresse IP du serveur FTP et le numéro du port d’écoute, respectivement.

3. Dans les champs **Username** et **Password**, saisissez les informations d’authentification.

4. Dans la liste **Connection Mode**, sélectionnez le mode de connexion au FTP que vous souhaitez utiliser, **Active** dans cet exemple.

**Placer des fichiers sur le serveur FTP**

Configurez le composant `tFTPPut` pour placer différents fichiers locaux dans le répertoire racine du serveur FTP.

**Procédure**

1. Double-cliquez sur le composant `tFTPPut` pour ouvrir sa vue Basic settings.
2. Spécifiez les informations de connexion requis pour accéder au serveur FTP. Dans cet exemple, cochez la case **Use an existing connection** et, dans la liste **Component list**, sélectionnez le composant de connexion pour réutiliser les informations de connexion précédemment définies.

3. Dans le champ **Local directory**, spécifiez le répertoire local contenant les fichiers à placer sur le serveur FTP. Dans cet exemple, spécifiez **D:/components**.

4. Dans le champ **Remote directory**, spécifiez le répertoire du serveur FTP dans lequel placer les fichiers. Dans cet exemple, saisissez **/**, ce qui indique le répertoire racine du serveur FTP.

5. Décrochez la case **Move to the current directory**.

6. Dans la table **Files**, cliquez deux fois sur le bouton **[+]** pour ajouter deux lignes et, dans les deux lignes de la colonne **Filemask**, saisissez *.txt* et *.png* respectivement, ce qui signifie que seuls les fichiers texte et les fichiers .png du répertoire local spécifié seront placés dans le répertoire racine du serveur FTP.

**Fermer la connexion au serveur FTP**

Configurez le composant **tFTPClose** pour fermer la connexion au serveur FTP.

**Procédure**

1. Double-cliquez sur le composant **tFTPClose** pour ouvrir sa vue **Basic settings**.

2. Dans la liste déroulante **Component list**, sélectionnez le composant **tFTPConnection** ouvrant la connexion que vous souhaitez fermer. Dans cet exemple, un seul **tFTPConnection** est utilisé, il est donc sélectionné par défaut.
Exécuter le Job pour placer des fichiers sur le serveur FTP

Après avoir configuré le Job et ses composants pour placer des fichiers sur le serveur FTP, vous pouvez exécuter le Job et vérifier ses résultats d’exécution.

Procédure

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder le Job et appuyez sur **F6** pour l’exécuter.
2. Connectez-vous au serveur FTP afin de vérifier les résultats.

Comme affiché ci-dessus, seuls les fichiers texte et .png du répertoire local sont placés sur le serveur FTP.
tFTPRename

Ce composant renomme des fichiers dans un répertoire FTP.

**Propriétés du tFTPRename Standard**

Ces propriétés sont utilisées pour configurer le tFTPRename s'exécutant dans le framework de Jobs Standard.

Le composant tFTPRename Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Property Type | Sélectionnez la manière de configurer les informations de connexion.  
  • *Built-In* : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.  
  • *Repository* : les informations de connexion stockées centralement dans le *Repository* > *Metadata* seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue *Repository Content*, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. |

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste *Component List* pour réutiliser les paramètres d'une connexion que vous avez déjà définie. |
| Host | Saisissez l'adresse IP ou le nom de l'hôte du serveur FTP. |
| Port | Saisissez le numéro du port d'écoute du serveur FTP. |
| Username et Password | Saisissez les données d'authentification de l'utilisateur au serveur FTP.  
  Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ *Password*, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Remote directory | Chemin d'accès au répertoire FTP dans lequel sont disponibles les fichiers à renommer. |
| Move to the current directory | Cochez cette case pour modifier le répertoire en l’un des répertoires spécifiés dans le champ *Remote directory*. Le composant FTP suivant dans le Job prendra ce répertoire comme racine du répertoire distant lors de l'utilisation de la même connexion. |
Cette propriété est disponible uniquement lorsque la case **Use an existing connection** est cochée.

| Overwrite file | Sélectionnez l'action à effectuer lorsque le fichier existe déjà.  
|                | • **never**: ne jamais écraser le fichier.  
|                | • **always**: toujours écraser le fichier.  
|                | • **size different**: écraser le fichier lorsque la taille du fichier est différente. |

| SFTP Support | Cochez cette case pour vous connecter au serveur FTP via une connexion SFTP. |

| Authentication method | Sélectionnez la méthode d'authentification SFTP, **Public key** ou **Password**.  
|                       | • **Public key**: saisissez le chemin d'accès à la clé privée et la phrase secrète pour la clé, dans les champs **Private key** et **Key Passphrase**, respectivement.  
|                       | • **Password**: saisissez le mot de passe requis.  
|                       | Cette propriété est disponible uniquement lorsque la case **SFTP Support** est cochée. |

| Filename encoding | Cochez cette case pour paramétrer l'encodage utilisé pour convertir les noms de fichiers de chaînes de caractères à octets. L'encodage utilisé doit être le même que celui utilisé sur le serveur SFTP. Si la version du serveur SFTP est supérieure à 3, l'encodage doit être **UTF-8**. Si ce n'est pas le cas, une erreur survient.  
|                   | Cette propriété est disponible uniquement lorsque la case **SFTP Support** est cochée. |

| FTPS Support | Cochez cette case pour vous connecter au serveur FTP via une connexion FTPS. |

| Keystore File | Saisissez le chemin d'accès au fichier Keystore (fichier protégé par un mot de passe et contenant plusieurs clés et certificats).  
|               | Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée. |

| Keystore Password | Saisissez votre mot de passe Keystore.  
|                  | Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée. |

| Security Mode | Sélectionnez le mode de sécurité dans la liste **Implicit** ou **Explicit**.  
|              | Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée. |

| Files | Spécifiez les fichiers à renommer et les nouveau noms.  
|       | • **Filemask**: spécifiez le fichier à renommer en saisissant le nom du fichier ou le masque du
fichier à l'aide de caractères de remplacement ou d'expressions régulières.
• **New name** : saisissez le nouveau nom du fichier.

| **Connection mode** | Sélectionnez dans la liste le mode de connexion, **Passive** ou **Active**. |
| **Encoding** | Sélectionnez dans la liste un type d’encodage ou sélectionnez **CUSTOM** et définissez-le manuellement. |
| **Die on error** | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.  
Décocochez la case pour ignorer les lignes en erreur et terminer le processus. |

**Advanced settings**

| **Use Socks Proxy** | Cochez cette case si vous utilisez un proxy et, dans les champs **Proxy host**, **Proxy port**, **Proxy user** et **Proxy password** qui s’affichent, spécifiez les paramètres du serveur proxy. |
| **Ignore Failure At Quit (FTP)** | Cochez cette case pour ignorer les erreurs de fermeture de bibliothèques ou les erreurs de fermeture FTP. |
| **Data Channel Protection Level** | Niveau de protection du canal de données avec lequel les données sont transférées entre le client et le serveur.  
Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée. |
Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Variables globales**

| **ERROR_MESSAGE** | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |
| **NB_FILE** | Nombre de fichiers traités. Cette variable est une variable After et retourne un entier. |
| **CURRENT_STATUS** | Résultat d’exécution du composant. Cette variable est une variable Flow et retourne une chaîne de caractères. |
Utilisation

| Règle d'utilisation | Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin. |

Renommer un fichier situé sur un serveur FTP

Voici un exemple d’utilisation des composants FTP de Talend pour renommer un fichier situé sur un serveur FTP.

Créer un Job pour renommer un fichier sur un serveur FTP

Créez un Job pour vous connecter à un serveur FTP, renommer un fichier sur le serveur et fermer la connexion au serveur.

Avant de commencer

**Prérequis** : Afin de reproduire ce scénario, un serveur FTP doit être démarré et un fichier doit être sur ce serveur. Dans cet exemple, le fichier *movies.json* est placé dans le dossier *movies*, sous le répertoire racine du serveur FTP.

<table>
<thead>
<tr>
<th>Remote site</th>
<th>/movies</th>
</tr>
</thead>
<tbody>
<tr>
<td>!movies.json</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Filename</th>
<th>Filesize</th>
<th>Filetype</th>
<th>Last modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>movies.json</td>
<td>299</td>
<td>JSON file</td>
<td>6/17/2013 4:43:01 PM</td>
</tr>
</tbody>
</table>
**Procédure**

1. Créez un nouveau Job et ajoutez un **tFTPConnection**, un **tFTPRename** et un **tFTPClose** en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le **tFTPConnection** au **tFTPRename** à l'aide d'un lien **Trigger > OnSubjobOk**.
3. Reliez le **tFTPRename** au **tFTPClose** à l'aide d'un lien **Trigger > OnSubjobOk**.

**Ouvrir une connexion au serveur FTP**

Configurez le **tFTPConnection** pour ouvrir une connexion au serveur FTP.

**Procédure**

1. Double-cliquez sur le **tFTPConnection** pour ouvrir sa vue **Basic settings**.
2. Dans les champs **Host** et **Port**, saisissez l'adresse IP du serveur FTP et le numéro du port d'écoute, respectivement.
3. Dans les champs **Username** et **Password**, saisissez les informations d'authentification.

**Renommer le fichier sur le serveur FTP**

Configurez le **tFTPRename** pour renommer le fichier sur le serveur FTP.

**Procédure**

1. Double-cliquez sur le composant **tFTPRename** pour ouvrir sa vue **Basic settings**.
2. Spécifiez les informations de connexion requises pour accéder au serveur FTP. Dans cet exemple, cochez la case **Use an existing connection** et, dans la liste déroulante **Component list**, sélectionnez le composant de connexion afin de réutiliser les informations de connexion précédemment définies.
3. Dans le champ **Remote directory**, saisissez le répertoire du serveur FTP où se situe le fichier à renommer. Dans cet exemple, saisissez **/movies**.
4. Décochez la case **Move to the current directory**.
5. Sous la table **Files**, cliquez sur le bouton [+] pour ajouter une ligne, puis saisissez le nom du fichier existant dans la colonne **Filemask**. Dans la colonne **New name**, saisissez le nouveau nom du fichier. Dans cet exemple, les noms sont respectivement **movies.json** et **action_movies.json**.
Fermer la connexion au serveur FTP

Configurez le tFTPClose pour fermer la connexion au serveur FTP.

Procédure

1. Double-cliquez sur le composant tFTPClose pour ouvrir sa vue Basic settings.

2. Dans la liste Component list, sélectionnez le composant tFTPConnection ouvrant la connexion que vous souhaitez fermer. Dans cet exemple, un seul tFTPConnection est utilisé, il est donc sélectionné par défaut.

Exécuter le Job pour renommer le fichier sur le serveur FTP

Après avoir configuré le Job et ses composants pour renommer le fichier sur le serveur FTP, vous pouvez exécuter le Job et vérifier ses résultats d’exécution.

Procédure

1. Appuyez sur les touches Ctrl + S afin de sauvegarder le Job, puis sur F6 pour l’exécuter.
2. Connectez-vous au serveur FTP afin de vérifier les résultats.

Comme affiché ci-dessus, le fichier sur le serveur FTP a été renommé de movies.json à action_movies.json.
tFTPTruncate

Ce composant tronque des fichiers dans un répertoire FTP.

Propriétés du tFTPTruncate Standard

Ces propriétés sont utilisées pour configurer le tFTPTruncate s’exécutant dans le framework de Jobs Standard.

Le composant tFTPTruncate Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |
| Host                     | Saisissez l’adresse IP ou le nom de l’hôte du serveur FTP. |

| Port                     | Saisissez le numéro du port d’écoute du serveur FTP. |

| Username et Password     | Saisissez les données d’authentification de l’utilisateur au serveur FTP. |
|                         | Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |

| Remote directory         | Chemin d’accès au répertoire FTP dans lequel tronquer les fichiers. |

| Move to the current directory | Cochez cette case pour modifier le répertoire en l’un des répertoires spécifiés dans le champ Remote directory. Le composant FTP suivant dans le Job prendra ce répertoire comme racine du répertoire distant lors de l’utilisation de la même connexion. |
| **tFTPTruncate** |  
| --- | --- |
| **Cette propriété est disponible uniquement lorsque la case Use an existing connection est cochée.** |  
| **SFTP Support** | Cochez cette case pour vous connecter au serveur FTP via une connexion SFTP. |
| **Authentication method** | Sélectionnez la méthode d’authentification SFTP, Public key ou Password.  
- **Public key** : saisissez le chemin d’accès à la clé privée et la phrase secrète pour la clé, dans les champs Private key et Key Passphrase, respectivement.  
- **Password** : saisissez le mot de passe requis.  
Cette propriété est disponible uniquement lorsque la case SFTP Support est cochée. |
| **Filename encoding** | Cochez cette case pour paramétrer l’encodage utilisé pour convertir les noms de fichiers de chaînes de caractères à octets. L’encodage utilisé doit être le même que celui utilisé sur le serveur SFTP. Si la version du serveur SFTP est supérieure à 3, l’encodage doit être UTF-8. Si ce n’est pas le cas, une erreur survient.  
Cette propriété est disponible uniquement lorsque la case SFTP Support est cochée. |
| **FTPS Support** | Cochez cette case pour vous connecter au serveur FTP via une connexion FTPS. |
| **Keystore File** | Saisissez le chemin d’accès au fichier Keystore (fichier protégé par un mot de passe et contenant plusieurs clés et certificats).  
Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée. |
| **Keystore Password** | Saisissez votre mot de passe Keystore.  
Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée. |
| **Security Mode** | Sélectionnez le mode de sécurité dans la liste Implicit ou Explicit.  
Cette propriété est disponible uniquement lorsque la case FTPS Support est cochée. |
| **Use Perl5 Regex Expression as Filemask** | Cochez cette case pour utiliser les expression régulières Perl5 dans le champ Files comme filtres de fichiers. Cela est utile lorsque le nom du fichier à traiter contient des caractères spéciaux, comme des parenthèses.  
Pour plus d’informations concernant la syntaxe des expressions régulières Perl5, consultez Perl5 Regular Expression Syntax (en anglais). |
| **Files** | Noms ou chemins des fichiers à tronquer. Vous pouvez spécifier plusieurs fichiers dans une ligne en saisissant des caractères de remplacement ou une expression régulière. |
**Connection mode**

Sélectionnez dans la liste le mode de connexion, **Passive** ou **Active**.

**Encoding**

Sélectionnez dans la liste un type d’encodage ou sélectionnez **CUSTOM** et définissez-le manuellement.

### Advanced settings

**Use Socks Proxy**

Cochez cette case si vous utilisez un proxy et, dans les champs **Proxy host**, **Proxy port**, **Proxy user** et **Proxy password** qui s’affichent, spécifiez les paramètres du serveur proxy.

**Ignore Failure At Quit (FTP)**

Cochez cette case pour ignorer les erreurs de fermeture de bibliothèques ou les erreurs de fermeture FTP.

**Data Channel Protection Level**

Niveau de protection du canal de données avec lequel les données sont transférées entre le client et le serveur. Pour plus d’informations, consultez [RFC 2228: FTP Security Extensions](en anglais). Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée.

**Protection Buffer Size**

Taille maximale (en octets) des blocs de données encodées à transférer entre le client et le serveur. Pour plus d’informations, consultez [RFC 2228: FTP Security Extensions](en anglais). Cette propriété est disponible uniquement lorsque la case **FTPS Support** est cochée.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

**ERROR_MESSAGE**

Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

**NB_FILE**

Nombre de fichiers traités. Cette variable est une variable After et retourne un entier.

**CURRENT_STATUS**

Résultat d’exécution du composant. Cette variable est une variable Flow et retourne une chaîne de caractères.

### Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé comme sous-Job a un seul composant mais il peut aussi être utilisé comme composant de sortie ou de fin.
**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tFuzzyMatch

Ce composant compare une colonne du flux principal avec une colonne du flux de référence et extrait les données du flux principal en affichant la distance.

Propriétés du tFuzzyMatch Standard

Ces propriétés sont utilisées pour configurer le tFuzzyMatch s’exécutant dans le framework de Jobs Standard.
Le composant tFuzzyMatch Standard appartient à la famille Data Quality.
Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.
Deux colonnes en lecture seule sont ajoutées automatiquement au schéma de sortie : Value et Match. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
| Matching type        | Sélectionnez l’algorithme de correspondance approprié parmi les suivants :
Levenshtein : Basé sur la théorie de la distance d’édition. Il calcule le nombre d’insertion, de suppression ou de substitution nécessaire pour correspondre à la référence.
Metaphone : Basé sur la phonétique. Tout d’abord, il charge les sonorités de toutes les entrées du flux de référence (Lookup) et les compare avec celles des entrées du flux principal (Main).
Double Metaphone : une nouvelle version de l’algorithme phonétique Metaphone, qui produit des résultats plus précis que l’algorithme original. Il peut retourner à la fois un code primaire et secondaire pour une chaîne de caractères. Pour éviter toute ambiguïté que le Metaphone n’aurait pas pu détecter, utilisez cette option. |
| Min distance         | (Levenshtein uniquement) Définissez le nombre minimum de modifications autorisées pour correspondre à la référence. Si vous définissez 0, seules les correspondances exactes sont extraites. |
### Max distance
(Levenshtein uniquement) Définissez le nombre maximum de modifications pouvant correspondre à la référence.

### Matching column
Sélectionnez la colonne du flux principal qui doit être comparée avec la colonne clé du flux de référence (Lookup).

### Unique matching
Cochez cette case pour obtenir la meilleure correspondance possible, si plusieurs correspondances sont disponibles.

### Matching item separator
Si plusieurs correspondances sont disponibles, elles seront toutes affichées sauf si la case Unique match est cochée. Définissez le séparateur entre chaque correspondance.

### Advanced settings
| tStatCatcher Statistics | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

### Global Variables
| Global Variables | NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.  
ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.  
Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.  
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation
| Règle d'utilisation | Ce composant n'est pas un composant de début (fond vert) et il nécessite deux composants d'entrée et un composant de sortie. |
Scénario 1 : Distance de Levenshtein de 0 pour les prénoms

Ce scénario décrit un Job constitué de quatre composants dont le but est de vérifier la distance d’édition comprise entre les colonnes First Name (prénom) d’un fichier d’entrée et de comparer les données avec celles du fichier de référence. Le résultat de cette vérification de la distance de Levenshtein et le contenu du flux principal sont affichés dans une table.

Construire le Job

Procédure

1. Cliquez et déposez les composants suivants de la Palette dans l’espace de modélisation : deux tFileInputDelimited, un tFuzzyMatch et un tLogRow.
2. Reliez le premier tFileInputDelimited au composant tFuzzyMatch à l’aide d’un lien Row > Main.
3. Connectez le second tFileInputDelimited au tFuzzyMatch à l’aide d’une connexion de type Row > Main (qui apparaît comme Lookup dans l’espace de modélisation graphique).
4. Reliez le tFuzzyMatch au composant de sortie tLogRow, à l’aide d’un lien Row > Main.

Configurer les composants

Procédure

2. Définissez le schéma du composant. Dans cet exemple, le schéma d’entrée possède deux colonnes, firstname et gender.
3. Configurez le second tFileInputDelimited de la même façon.

Avertissement :

Assurez-vous que la colonne de référence a bien été définie comme colonne clé dans le schéma du flux de référence (lookup).
4. Double-cliquez sur le tFuzzyMatch pour ouvrir sa vue Basic settings et vérifiez son schéma. Le schéma doit correspondre à celui du flux d’entrée principal (Main) afin que le flux principal soit comparé au flux de référence.

5. Sélectionnez la méthode à utiliser pour la vérification des données entrantes. Dans ce scénario, la correspondance à sélectionner dans le champ Matching type est de type Levenshtein.

6. Ensuite définissez la distance. Dans cette méthode, la distance est le nombre de caractères modifiés (insertion, suppression ou substitution) pour que l’entrée principale corresponde exactement à l’entrée de référence.
Dans ce scénario, configurez la distance à 0 au minimum et au maximum dans les champs Min. distance et Max. distance. Ce qui signifie que seule une correspondance exacte sera redirigée en flux de sortie.

7. Décochez la case Case sensitive pour ne pas prendre en compte la casse.

8. Vérifiez que la colonne de correspondance et la colonne de référence sont bien sélectionnées.

9. Laissez les autres paramètres par défaut.

**Exécuter le Job**

**Procédure**

Enregistrez le Job et appuyez sur la touche F6 pour exécuter le Job.

**Résultats**

Comme la distance d’édition définie est de 0 (min. et max.), la sortie correspond à une jointure entre le flux principal et le flux de référence (lookup), ainsi seules les correspondances totales avec une valeur égale à 0 sont affichées.

Pour un exemple plus évident avec une distance minimum de 1 et une distance maximum de 2, consultez Procédure à la page 1334

**Scénario 2 : Distance de Levenshtein de 1 ou 2 pour les prénoms**

Ce scénario est basé sur le scénario décrit ci-dessus. Seuls les paramètres distance minimum et maximum du composant tFuzzyMatch sont modifiés, ce qui modifiera la sortie affichée.

**Procédure**

1. Dans la vue Component du tFuzzyMatch, modifiez la distance minimum de 0 à 1. Ceci exclut directement les correspondances exactes (qui ont une distance de 0).
2. Maintenant mettez une distance maximum de 2. La sortie correspond à toutes les données mises en correspondance contenant une différence de 2 caractères au plus.

![tFuzzyMatch_1](image)

Aucune autre modification de paramètres n’est nécessaire.

3. Assurez-vous que le séparateur de champs (Matching item separator) est bien défini, vu que plusieurs entrées de référence peuvent correspondre à l’entrée du flux principal.

4. Enregistrez votre nouveau Job et appuyez sur F6 pour l’exécuter.

### Résultats

![Execution](image)

Etant donné que la distance d’édition a été définie à 2, certaines entrées du flux principal correspondent à plusieurs entrées de référence (lookup).

Vous pouvez utiliser une autre méthode, Métaphone, pour évaluer la distance entre le flux principal et le flux de référence, décrite dans le scénario suivant.

### Scénario 3 : Distance métaphonique pour les prénoms

Ce scénario est basé sur un scénario décrit plus haut.
Procédure

1. Modifiez le champ **Matching type** avec l’option **Metaphone**. Aucune distance minimum et aucune distance maximum ne sont à définir puisque cette méthode est basée sur les différences de phonétique avec la référence.

2. Enregistrez le Job et appuyez sur **F6**. La valeur phonétique est affichée avec les correspondances possibles.

Résultats
tGoogleDataprocManage

Ce composant crée ou supprime un cluster Dataproc dans Google Cloud Platform.

Propriétés du tGoogleDataprocManage Standard

Ces propriétés sont utilisées pour configurer le tGoogleDataprocManage s'exécutant dans le framework de Jobs Standard.

Le composant tGoogleDataprocManage Standard appartient à la famille Cloud.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project identifier</td>
<td>Saisissez l'ID de votre projet Google Cloud Platform. Si vous n'êtes pas certain de l'ID de votre projet, vérifiez dans la page Manage Resources de vos services Google Cloud Platform.</td>
</tr>
<tr>
<td>Cluster identifier</td>
<td>Saisissez l'ID de votre cluster Dataproc à utiliser.</td>
</tr>
<tr>
<td>Action</td>
<td>Sélectionnez l’action à effectuer sur votre cluster par le tGoogleDataprocManage.</td>
</tr>
<tr>
<td>• Start</td>
<td>pour créer un cluster,</td>
</tr>
<tr>
<td>• Stop</td>
<td>pour supprimer un cluster.</td>
</tr>
<tr>
<td>Version</td>
<td>Sélectionnez la version de l’image à utiliser pour créer un cluster Dataproc.</td>
</tr>
<tr>
<td>Zone</td>
<td>Sélectionnez la zone géographique dans laquelle sont utilisées les ressources de calcul et dans laquelle sont stockées et traitées les données.</td>
</tr>
<tr>
<td>Instance configuration</td>
<td>Saisissez les paramètres afin de déterminer combien d’instances master et worker doivent être utilisées par le cluster Dataproc à créer et configurez les performances de ces instances master et worker.</td>
</tr>
</tbody>
</table>
### Advanced settings

<table>
<thead>
<tr>
<th><strong>Wait for cluster ready</strong></th>
<th>Cochez cette case pour que le composant continue à s'exécuter jusqu'à ce que le cluster soit complètement configuré. Lorsque vous décochez cette case, ce composant arrête son exécution immédiatement après avoir envoyé la commande de création.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Master disk size</strong></td>
<td>Saisissez un nombre, sans guillemet, afin de déterminer la taille du disque de chaque instance maître.</td>
</tr>
<tr>
<td><strong>Master local SSD</strong></td>
<td>Saisissez un nombre, sans guillemet, afin de déterminer le nombre d'appareils de stockage SSD (solid-state drive) locaux à ajouter à chaque instance maître. Selon Google, ces SSD locaux conviennent uniquement aux stockages temporaires, comme les caches, pour les espaces de traitement ou les données à faible valeur. Il est recommandé d'utiliser les options de stockage durable de Google pour stocker les données importantes. Pour plus d'informations concernant les options de stockage de Google, consultez Durable storage options (en anglais).</td>
</tr>
<tr>
<td><strong>Worker disk size</strong></td>
<td>Saisissez un nombre, sans guillemet, afin de déterminer la taille du disque de chaque instance worker.</td>
</tr>
<tr>
<td><strong>Worker local SSD</strong></td>
<td>Saisissez un nombre, sans guillemet, afin de déterminer le nombre d'appareils de stockage SSD (solid-state drive) locaux à ajouter à chaque instance worker. Selon Google, ces SSD locaux conviennent uniquement aux stockages temporaires, comme les caches, pour les espaces de traitement ou les données à faible valeur. Il est recommandé d'utiliser les options de stockage durable de Google pour stocker les données importantes. Pour plus d'informations concernant les options de stockage de Google, consultez Durable storage options (en anglais).</td>
</tr>
<tr>
<td><strong>Network et Subnetwork</strong></td>
<td>Cochez une case ou l’autre pour utiliser respectivement un réseau ou sous-réseau Google Compute Engine pour le cluster à créer, pour activer les communications intra-cluster. Comme Google ne permet pas l’utilisation simultanée du réseau et du sous-réseau, cocher une case masque l’autre. Pour plus d’informations concernant la configuration du réseau pour les clusters Google Dataproc, consultez Dataproc Network (en anglais).</td>
</tr>
</tbody>
</table>
| **Initialization action**  | Dans cette table, sélectionnez les actions d’initialisation disponibles dans le bucket partagé dans Google Cloud Storage à exécuter sur tous les nœuds de votre cluster Dataproc, immédiatement après la configuration du cluster. Si vous devez utiliser des scripts d’initialisation personnalisés, chargez-les dans ce bucket Google partagé, afin que le tGoogleDataprocManage puisse les lire.  
  - Dans la colonne **Executable timeout**, saisissez entre guillemets doubles la période de temps déterminant la durée de l’exécution. Si l’exécutable n’est pas terminé à la fin de ce délai, un message d’erreur explicatif est retourné. La valeur est une chaîne de caractères contenant jusqu’à neuf chiffres après la virgule, par exemple ”3.5s” pour 3,5 secondes. Pour plus d’informations concernant ce bucket partagé et les actions d’initialisation, consultez Initialization actions (en anglais). |
| **tStatCatcher Statistics**| Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |
## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en standalone dans un sous-Job.</th>
</tr>
</thead>
</table>


tGoogleDriveConnection

Ce composant ouvre une connexion à Google Drive pouvant être réutilisée par d’autres composants Google Drive.

Propriétés du tGoogleDriveConnection Standard

Ces propriétés sont utilisées pour configurer le tGoogleDriveConnection s’exécutant dans le framework de Jobs Standard.

Le composant tGoogleDriveConnection Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong>: les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong>: les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton […] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Nom de l’application requis par Google Drive pour obtenir l’accès à ses API.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>OAuth Method</th>
<th>Sélectionnez une méthode OAuth utilisée pour accéder à Google Drive dans la liste déroulante.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Access Token (deprecated)</strong> : utilise un jeton d’accès pour accéder à Google Drive.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Installed Application (Id &amp; Secret)</strong> : utilise l’ID et le secret du client créés via la Console Google API pour accéder à Google Drive. Pour plus d’informations concernant cette méthode, consultez Google Identity Platform &gt; Installed applications (en anglais).</td>
</tr>
<tr>
<td></td>
<td>• <strong>Installed Application (JSON)</strong> : utilise un fichier JSON de secret, créé via la Console Google API contenant l’ID du client, son secret et d’autres paramètres OAuth 2.0 pour accéder à Google Drive.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Service Account</strong> : utilise un fichier JSON de compte de service, créé via la Console Google API, pour accéder à Google Drive. Pour plus d’informations concernant cette méthode, consultez Google Identity Platform &gt; Service accounts (en anglais).</td>
</tr>
</tbody>
</table>
Pour des informations plus détaillées relatives à l’accès à Google Drive via chaque méthode, consultez Méthodes OAuth pour accéder à Google Drive à la page 1342.

| Access Token       | Jeton d’accès généré via Google Developers OAuth 2.0 Playground.
|                    | Cette propriété est disponible uniquement lorsque l’option Access Token est sélectionnée dans la liste déroulante OAuth Method. |
| Client ID et Client Secret | ID et secret du client.
|                    | Ces deux propriétés sont disponibles uniquement lorsque l’option Installed Application (Id & Secret) est sélectionnée dans la liste déroulante OAuth Method. |
| Client Secret JSON | Chemin d’accès au fichier JSON contenant le secret du client.
|                    | Cette propriété est disponible uniquement lorsque l’option Installed Application (JSON) est sélectionnée dans la liste déroulante OAuth Method. |
| Service Account JSON | Chemin d’accès au fichier JSON du compte de service.
|                    | Cette propriété est disponible uniquement lorsque l’option Service Account est sélectionnée dans la liste déroulante OAuth Method. |
| Use Proxy          | Cochez cette case, lorsque vous travaillez derrière un proxy. Lorsque cette case est cochée, vous devez spécifier la valeur des paramètres suivants :
|                    | • Host : adresse IP du serveur du proxy HTTP.
|                    | • Port : numéro du port du serveur du proxy HTTP. |
| Use SSL            | Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive. Une fois cette case cochée, vous devez spécifier la valeur des paramètres suivants :
|                    | • Algorithm : nom de l’algorithme de chiffrement SSL.
|                    | • Keystore File : chemin d’accès au fichier TrustStore du certificat contenant la liste des certificat auxquels le client fait confiance.
|                    | • Password : mot de passe utilisé pour vérifier l’intégrité des données TrustStore. |

**Advanced settings**

| DataStore Path | Chemin d’accès au fichier Credential contenant le jeton de renouvellement.

**Remarque** : Lorsque l’ID du client, le secret du client ou tout autre paramètre relatif à l’authentification de l’application installée change, vous devez supprimer ce fichier Credential manuellement, avant d’exécuter à nouveau votre Job.
Cette propriété est disponible uniquement lorsque l’option **Installed Application (Id & Secret)** ou **Installed Application (JSON)** est sélectionnée dans la liste déroulante **OAuth Method**.

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Google Drive. Dans un Job, il est généralement utilisé pour ouvrir une connexion à Google Drive pouvant être réutilisée par d’autres composants. |

**Méthodes OAuth pour accéder à Google Drive**

Talend fournit les quatre méthodes OAuth suivantes pour accéder à Google Drive à l’aide des composants et de l’assistant de métadonnées Google Drive.

- Application installée (ID et Secret)
- Application installée (JSON)
- Compte de service
- Jeton d’accès (déprécié)

**Accéder à Google Drive à l’aide de l’ID et du Secret du client**

Pour utiliser l’ID et le secret du client pour accéder à Google Drive, vous devez d’abord générer l’ID et le secret du client en effectuant les étapes suivantes à l’aide de Google Chrome.

**Avant de commencer**

Un compte Google doit avoir été créé pour utiliser Google Drive.

**Procédure**


2. Allez dans la page Library. Dans le panneau de droite, trouvez Google Drive API et activez l'API. Google Drive API vous permettant d'accéder aux ressources depuis Google Drive.
3. Allez dans la page Credentials, cliquez sur OAuth consent screen dans le panneau de droite et configurez un nom de produit dans le champ Product name shown to users. Dans cet exemple, saisissez TalendProduct. Cela fait, cliquez sur Save.
5. Cliquez sur **Create**. Votre ID et votre Secret s'affichent. Les composants et l'assistant de métadonnées Google Drive peuvent les utiliser pour accéder à Google Drive à l'aide de la méthode OAuth **Installed Application (Id & Secret)**.

**Accéder à Google Drive à l'aide d'un fichier JSON contenant le secret de client**

Pour utiliser un fichier JSON contenant un secret de client pour accéder à Google Drive, vous devez d'abord télécharger le fichier JSON depuis la Console de Google API en effectuant les étapes suivantes, dans Google Chrome.
Avant de commencer

L'ID et le secret du client doivent avoir été créés dans la Console de Google API. Pour plus d'informations, consultez Accéder à Google Drive à l'aide de l'ID et du Secret du client à la page 1342.

Procédure

1. Allez à la page Google API Console.
2. Allez à la page Credentials.
3. Cliquez sur le bouton Download JSON pour télécharger le fichier JSON contenant le secret du client et le stocker de manière sécurisée dans un dossier local. Ce fichier JSON peut être utilisé par les composants et l’assistant de métadonnées Google Drive, pour accéder à Google Drive via la méthode OAuth Installed Application (JSON).

Accéder à Google Drive à l'aide d'un fichier JSON contenant un compte de service

Pour utiliser un fichier JSON contenant un compte de service pour accéder à Google Drive, vous devez d'abord créer un compte de service dans la Console Google API, puis télécharger le fichier JSON en effectuant les étapes suivantes dans Google Chrome.

Avant de commencer

1. Un compte Google doit avoir été créé pour utiliser Google Drive.
2. Dans la Console Google API, votre projet doit avoir été créé, l'API Google Drive doit avoir été activée et le nom du produit configuré. Pour plus d'informations concernant ces configurations, consultez Accéder à Google Drive à l'aide de l'ID et du Secret du client à la page 1342.

Procédure

1. Allez à la page Google API Console.
2. Ouvrez la page Service accounts. Si cela vous est demandé, sélectionnez votre projet.
3. Cliquez sur **CREATE SERVICE ACCOUNT**.
4. Dans la fenêtre **Create service account**, saisissez un nom pour le compte de service, cochez la case **Furnish a new private key** et sélectionnez le type de clé, **JSON**.
5. Cliquez sur **Create**. Dans la fenêtre qui s’ouvre, sélectionnez un dossier et cliquez sur **Save** pour stocker votre fichier JSON de manière sécurisée. Ce fichier JSON peut être utilisé par les composants et l’assistant de métadonnées Google Drive pour accéder à Google Drive via la méthode OAuth Service Account.

**Accéder à Google Drive à l’aide d’un jeton d’accès (déprécié)**

Pour utiliser un jeton d’accès pour accéder à Google Drive, vous devez d’abord générer le jeton d’accès en effectuant les étapes suivantes, à l’aide de Google Developers OAuth Playground.

**Avant de commencer**

1. Un compte Google doit avoir été créé pour utiliser Google Drive.

2. L’ID et le secret du client doivent avoir été créés dans la Console Google API. Pour plus d’informations, consultez **Accéder à Google Drive à l’aide de l’ID et du Secret du client** à la page 1342.

**Procédure**

1. Allez à la page Google Developers OAuth Playground.

2. Cliquez sur **OAuth 2.0 Configuration** et cochez la case **Use your own OAuth credentials**. Dans les champs **OAuth Client ID** et **OAuth Client secret** en dessous, saisissez respectivement l’ID et le secret du client, précédemment créés.
3. Dans l’étape OAuth 2.0 Playground Step 1, sélectionnez https://www.googleapis.com/auth/drive, sous Drive API v3, pour l’API Google Drive API et cliquez sur Authorize APIs, puis cliquez sur Allow, afin de générer le code d’autorisation.

Le jeton d’accès OAuth s’affiche dans le panneau de droite, comme dans l’image ci-dessous. Il peut être utilisé par les composants et l’assistant de métadonnées Google Drive pour accéder à Google Drive via la méthode OAuth Access Token.
Notez que ce jeton d’accès expire au bout de 3 600 secondes. Vous pouvez cliquer sur **Refresh access token** dans l’étape **OAuth 2.0 Playground Step 2**, afin de le rafraîchir.

**Scénario associé**

**Gestion des fichiers avec Google Drive** à la page 1374
tGoogleDriveCopy

Ce composant crée une copie d’un fichier/dossier dans Google Drive.

Propriétés du tGoogleDriveCopy Standard

Ces propriétés sont utilisées pour configurer le tGoogleDriveCopy s’exécutant dans le framework de Jobs Standard.
Le composant tGoogleDriveCopy Standard appartient à la famille Cloud.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>Repository</td>
<td>les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

Connection Component  | Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant. |

Application Name  | Nom de l’application requis par Google Drive pour obtenir l’accès à ses API. |

OAuth Method  | Sélectionnez une méthode OAuth utilisée pour accéder à Google Drive dans la liste déroulante. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Token (deprecated)</td>
<td>utilise un jeton d’accès pour accéder à Google Drive.</td>
</tr>
<tr>
<td>Installed Application (Id &amp; Secret)</td>
<td>utilise l’ID et le secret du client créés via la Console Google API pour accéder à Google Drive. Pour plus d’informations concernant cette méthode, consultez Google Identity Platform &gt; Installed applications (en anglais).</td>
</tr>
<tr>
<td>Installed Application (JSON)</td>
<td>utilise un fichier JSON de secret, créé via la Console Google API contenant l’ID du client, son secret et d’autres paramètres OAuth 2.0 pour accéder à Google Drive.</td>
</tr>
<tr>
<td>Client ID et Client Secret</td>
<td>ID et secret du client. Ces deux propriétés sont disponibles uniquement lorsque l’option Installed Application (Id &amp; Secret) est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td>Client Secret JSON</td>
<td>Chemin d’accès au fichier JSON contenant le secret du client. Cette propriété est disponible uniquement lorsque l’option Installed Application (JSON) est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td>Service Account JSON</td>
<td>Chemin d’accès au fichier JSON du compte de service. Cette propriété est disponible uniquement lorsque l’option Service Account est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
</tbody>
</table>
| Use Proxy | Cochez cette case, lorsque vous travaillez derrière un proxy. Lorsque cette case est cochée, vous devez spécifier la valeur des paramètres suivants :
  - **Host** : adresse IP du serveur du proxy HTTP.
  - **Port** : numéro du port du serveur du proxy HTTP. |
| Use SSL | Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive. Une fois cette case cochée, vous devez spécifier la valeur des paramètres suivants :
  - **Algorithm** : nom de l’algorithme de chiffrement SSL.
  - **Keystore File** : chemin d’accès au fichier TrustStore du certificat contenant la liste des certificats auxquels le client fait confiance.
  - **Password** : mot de passe utilisé pour vérifier l’intégrité des données TrustStore. |
| Copy Mode | Sélectionnez type de l’élément à copier.
  - **File** : sélectionnez cette option lorsque vous devez copier un fichier. |
**Folder** : sélectionnez cette option lorsque vous devez copier un dossier.

**Source**
Nom ou ID du fichier/dossier source à copier.

**Source Access Mode**
 Sélectionnez la méthode par laquelle est spécifié le fichier/dossier source, by Name or by Id.

**Destination Folder Name**
Nom ou ID du dossier de destination dans lequel la copie du fichier/dossier source sera sauvegardée.

**Destination Access Mode**
Sélectionnez la méthode par laquelle le dossier de destination est spécifié, by Name ou by Id.

**Rename (set new title)**
Cochez cette case afin de renommer la copie du fichier ou dossier dans le dossier de destination. Dans le champ Destination Name qui s’affiche, saisissez le nom du fichier/dossier après copie dans son dossier de destination.

**Remove Source File**
Cochez cette case afin de supprimer le fichier source une fois copié dans le dossier de destination.
Cette propriété est disponible uniquement lorsque l’option File est sélectionnée dans la liste déroulante.

**Schema et Edit schema**
Un schéma est une description de lignes et définit les champs à traiter et à passer au composant suivant.
Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton [...] à côté de Edit schema afin de voir le schéma prédéfini contenant les champs suivants :
- **sourceID** : ID du fichier/dossier source.
- **destinationID** : ID du fichier/dossier de destination.

**Advanced settings**

**DataStore Path**
Chemin d’accès au fichier Credential contenant le jeton de renouvellement.

**Remarque** : Lorsque l’ID du client, le secret du client ou tout autre paramètre relatif à l’authentification de l’application installée change, vous devez supprimer ce fichier Credential manuellement, avant d’exécuter à nouveau votre Job.
Cette propriété est disponible uniquement lorsque l’option Installed Application (Id & Secret) ou Installed Application (JSON) est sélectionnée dans la liste déroulante OAuth Method.

**tStatCatcher Statistics**
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.
### Variables globales

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>SOURCE_ID</td>
<td>ID du fichier/dossier source. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>DESTINATION_ID</td>
<td>ID du fichier/dossier de destination. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce composant peut être utilisé en standalone ou en tant que composant de début dans un Job ou un sous-job.</td>
</tr>
</tbody>
</table>

### Scénario associé

Gestion des fichiers avec Google Drive à la page 1374
tGoogleDriveCreate

Ce composant crée un nouveau dossier dans Google Drive.

Propriétés du tGoogleDriveCreate Standard

Ces propriétés sont utilisées pour configurer le tGoogleDriveCreate s’exécutant dans le framework de Jobs Standard.

Le composant tGoogleDriveCreate Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste <strong>Connection Component</strong>.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant. |

| Application Name | Nom de l’application requis par Google Drive pour obtenir l’accès à ses API. |

<table>
<thead>
<tr>
<th>OAuth Method</th>
<th>Sélectionnez une méthode OAuth utilisée pour accéder à Google Drive dans la liste déroulante.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Access Token (deprecated)</strong> : utilise un jeton d’accès pour accéder à Google Drive.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Installed Application (Id &amp; Secret)</strong> : utilise l’ID et le secret du client créés via la Console Google API pour accéder à Google Drive. Pour plus d’informations concernant cette méthode, consultez [Google Identity Platform &gt; Installed applications](en anglais).</td>
</tr>
<tr>
<td></td>
<td>• <strong>Installed Application (JSON)</strong> : utilise un fichier JSON de secret, créé via la Console Google API contenant l’ID du client, son secret et d’autres paramètres OAuth 2.0 pour accéder à Google Drive.</td>
</tr>
<tr>
<td>Access Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Service Account</strong></td>
<td>utilise un fichier JSON de compte de service, créé via la Console Google API, pour accéder à Google Drive. Pour plus d’informations concernant cette méthode, consultez [Google Identity Platform &gt; Service accounts](en anglais). Pour des informations plus détaillées relatives à l’accès à Google Drive via chaque méthode, consultez <a href="#">Méthodes OAuth pour accéder à Google Drive</a> à la page 1342.</td>
</tr>
</tbody>
</table>

**Access Token**

Jeton d’accès généré via Google Developers OAuth 2.0 Playground.

Cette propriété est disponible uniquement lorsque l’option Access Token est sélectionnée dans la liste déroulante OAuth Method.

**Client ID et Client Secret**

ID et secret du client.

Ces deux propriétés sont disponibles uniquement lorsque l’option Installed Application (Id & Secret) est sélectionnée dans la liste déroulante OAuth Method.

**Client Secret JSON**

Chemin d’accès au fichier JSON contenant le secret du client.

Cette propriété est disponible uniquement lorsque l’option Installed Application (JSON) est sélectionnée dans la liste déroulante OAuth Method.

**Service Account JSON**

Chemin d’accès au fichier JSON du compte de service.

Cette propriété est disponible uniquement lorsque l’option Service Account est sélectionnée dans la liste déroulante OAuth Method.

**Use Proxy**

Cochez cette case, lorsque vous travaillez derrière un proxy. Lorsque cette case est cochée, vous devez spécifier la valeur des paramètres suivants :

- **Host** : adresse IP du serveur du proxy HTTP.
- **Port** : numéro du port du serveur du proxy HTTP.

**Use SSL**

Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive. Une fois cette case cochée, vous devez spécifier la valeur des paramètres suivants :

- **Algorithm** : nom de l’algorithme de chiffrement SSL.
- **Keystore File** : chemin d’accès au fichier TrustStore du certificat contenant la liste des certificat auxquels le client fait confiance.
- **Password** : mot de passe utilisé pour vérifier l’intégrité des données TrustStore.

**Parent Folder Name**

Saisissez le nom ou l’ID du dossier parent dans lequel créer le nouveau dossier.

**Access Method**

Sélectionnez la méthode par laquelle le dossier parent est spécifié, by Name ou by Id.
<table>
<thead>
<tr>
<th><strong>New Folder Name</strong></th>
<th>Saisissez le nom du nouveau dossier à créer.</th>
</tr>
</thead>
</table>
| **Schema et Edit schema** | Un schéma est une description de lignes et définit les champs à traiter et à passer au composant suivant. Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton [...] à côté de **Edit schema** afin de voir le schéma prédéfini contenant les champs suivants :  
  - **parentFolderId** : ID du dossier parent.  
  - **newFolderId** : ID du nouveau dossier. |

**Advanced settings**

| **DataStore Path** | Chemin d'accès au fichier Credential contenant le jeton de renouvellement.  
  ❗️ **Remarque** : Lorsque l'ID du client, le secret du client ou tout autre paramètre relatif à l'authentification de l'application installée change, vous devez supprimer ce fichier Credential manuellement, avant d'exécuter à nouveau votre Job.  
  Cette propriété est disponible uniquement lorsque l'option **Installed Application (Id & Secret)** ou **Installed Application (JSON)** est sélectionnée dans la liste déroulante **OAuth Method**. |
|---------------------|---------------------------------------------|

| **tStatCatcher Statistics** | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. |

**Variables globales**

<table>
<thead>
<tr>
<th><strong>ERROR_MESSAGE</strong></th>
<th>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PARENT_FOLDER_ID</strong></td>
<td>ID du dossier parent. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>NEW_FOLDER_ID</strong></td>
<td>ID du nouveau dossier. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

| **Règle d’utilisation** | Ce composant peut être utilisé en standalone ou en tant que composant de début dans un Job ou un sous-job. |

**Scénario associé**

Gestion des fichiers avec Google Drive à la page 1374
tGoogleDriveDelete

Ce composant supprime un fichier ou un dossier dans Google Drive.

**Propriétés du tGoogleDriveDelete Standard**

Ces propriétés sont utilisées pour configurer le tGoogleDriveDelete s'exécutant dans le framework de Jobs Standard.

Le composant tGoogleDriveDelete Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n'est pas disponible lorsqu'un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant. |

| Application Name    | Nom de l’application requis par Google Drive pour obtenir l’accès à ses API. |

<table>
<thead>
<tr>
<th>OAuth Method</th>
<th>Sélectionnez une méthode OAuth utilisée pour accéder à Google Drive dans la liste déroulante.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Access Token (deprecated)</strong> : utilise un jeton d’accès pour accéder à Google Drive.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Installed Application (Id &amp; Secret)</strong> : utilise l’ID et le secret du client créés via la Console Google API pour accéder à Google Drive. Pour plus d’informations concernant cette méthode, consultez Google Identity Platform &gt; Installed applications (en anglais).</td>
</tr>
<tr>
<td></td>
<td>• <strong>Installed Application (JSON)</strong> : utilise un fichier JSON de secret, créé via la Console Google API contenant l’ID du client, son secret et d’autres paramètres OAuth 2.0 pour accéder à Google Drive.</td>
</tr>
</tbody>
</table>

Pour des informations plus détaillées relatives à l’accès à Google Drive via chaque méthode, consultez [Méthodes OAuth pour accéder à Google Drive](#) à la page 1342.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Client ID et Client Secret</td>
<td>ID et secret du client. Ces deux propriétés sont disponibles uniquement lorsque l’option Installed Application (Id &amp; Secret) est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td>Client Secret JSON</td>
<td>Chemin d’accès au fichier JSON contenant le secret du client. Cette propriété est disponible uniquement lorsque l’option Installed Application (JSON) est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td>Service Account JSON</td>
<td>Chemin d’accès au fichier JSON du compte de service. Cette propriété est disponible uniquement lorsque l’option Service Account est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
</tbody>
</table>
| Use Proxy | Cochez cette case, lorsque vous travaillez derrière un proxy. Lorsque cette case est cochée, vous devez spécifier la valeur des paramètres suivants :
- **Host** : adresse IP du serveur du proxy HTTP.
- **Port** : numéro du port du serveur du proxy HTTP. |
| Use SSL | Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive. Une fois cette case cochée, vous devez spécifier la valeur des paramètres suivants :
- **Algorithm** : nom de l’algorithme de chiffrement SSL.
- **Keystore File** : chemin d’accès au fichier TrustStore du certificat contenant la liste des certificat auxquels le client fait confiance.
- **Password** : mot de passe utilisé pour vérifier l’intégrité des données TrustStore. |
| File/Folder | Saisissez le nom ou l’ID du fichier/dossier à supprimer. |
| Delete Mode | Sélectionnez la méthode par laquelle spécifier le fichier/dossier, by Name ou by Id. |
### Use Trash
Cochez cette case afin de déplacer le fichier/dossier à supprimer dans la corbeille.

### Schema et Edit schema
Un schéma est une description de lignes et définit les champs à traiter et à passer au composant suivant.
Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton [...] à côté du champ **Edit schema** pour voir le schéma prédéfini contenant une seule ligne, nommée **fileId**, décrivant l’ID du fichier/dossier.

### Advanced settings

#### DataStore Path
Chemins d’accès au fichier Credential contenant le jeton de renouvellement.

**Remarque** : Lorsque l’ID du client, le secret du client ou tout autre paramètre relatif à l’authentification de l’application installée change, vous devez supprimer ce fichier Credential manuellement, avant d’exécuter à nouveau votre Job.

Cette propriété est disponible uniquement lorsque l’option **Installed Application (Id & Secret)** ou **Installed Application (JSON)** est sélectionnée dans la liste déroulante **OAuth Method**.

#### tStatCatcher Statistics
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.

### Variables globales

#### ERROR_MESSAGE
Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

#### FILE_ID
ID du fichier/dossier. Cette variable est une variable After et retourne une chaîne de caractères.

### Utilisation

**Règle d’utilisation**
Ce composant peut être utilisé en standalone ou en tant que composant de début dans un Job ou un sous-job.

### Scénario associé
Aucun scénario n’est disponible pour ce composant.
tGoogleDriveGet

Ce composant récupère le contenu d'un fichier et télécharge le fichier dans un répertoire local.

Propriétés du tGoogleDriveGet Standard

Ces propriétés sont utilisées pour configurer le tGoogleDriveGet s'exécutant dans le framework de Jobs Standard.
Le composant tGoogleDriveGet Standard appartient à la famille Cloud.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>Repository</td>
<td>les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant. |

| Application Name    | Nom de l’application requis par Google Drive pour obtenir l’accès à ses API. |

<table>
<thead>
<tr>
<th>OAuth Method</th>
<th>Sélectionnez une méthode OAuth utilisée pour accéder à Google Drive dans la liste déroulante.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Token (deprecated)</td>
<td>utilise un jeton d'accès pour accéder à Google Drive.</td>
</tr>
<tr>
<td>Installed Application (Id &amp; Secret)</td>
<td>utilise l'ID et le secret du client créés via la Console Google API pour accéder à Google Drive. Pour plus d'informations concernant cette méthode, consultez Google Identity Platform &gt; Installed applications (en anglais).</td>
</tr>
<tr>
<td>Installed Application (JSON)</td>
<td>utilise un fichier JSON de secret, créé via la Console Google API contenant l'ID du client, son secret et d'autres paramètres OAuth 2.0 pour accéder à Google Drive.</td>
</tr>
<tr>
<td><strong>Access Token</strong></td>
<td>Jeton d'accès généré via Google Developers OAuth 2.0 Playground. Cette propriété est disponible uniquement lorsque l'option Access Token est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td><strong>Client ID et Client Secret</strong></td>
<td>ID et secret du client. Ces deux propriétés sont disponibles uniquement lorsque l'option Installed Application (Id &amp; Secret) est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td><strong>Client Secret JSON</strong></td>
<td>Chemin d'accès au fichier JSON contenant le secret du client. Cette propriété est disponible uniquement lorsque l'option Installed Application (JSON) est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td><strong>Service Account JSON</strong></td>
<td>Chemin d'accès au fichier JSON du compte de service. Cette propriété est disponible uniquement lorsque l'option Service Account est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
</tbody>
</table>
| **Use Proxy** | Cochez cette case, lorsque vous travaillez derrière un proxy. Lorsque cette case est cochée, vous devez spécifier la valeur des paramètres suivants :
  - **Host** : adresse IP du serveur du proxy HTTP.
  - **Port** : numéro du port du serveur du proxy HTTP. |
| **Use SSL** | Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive. Une fois cette case cochée, vous devez spécifier la valeur des paramètres suivants :
  - **Algorithm** : nom de l'algorithme de chiffrement SSL.
  - **Keystore File** : chemin d'accès au fichier TrustStore du certificat contenant la liste des certificats auxquels le client fait confiance.
  - **Password** : mot de passe utilisé pour vérifier l'intégrité des données TrustStore. |
| **File** | Saisissez le nom ou l'ID du fichier à télécharger. |
| **Access Method** | Sélectionnez la méthode par laquelle le fichier à télécharger est spécifié, by Name ou by Id. |
**Save As File**

Cochez cette case afin de sauvegarder le fichier dans un répertoire local. Dans le champ **Save to** qui s’affiche, parcourez votre système ou saisissez le chemin où sauvegarder le fichier à télécharger.

**Schema et Edit schema**

Un schéma est une description de lignes et définit les champs à traiter et à passer au composant suivant.

Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton [...] à côté de Edit schema afin de voir le schéma prédéfini ayant un champ nommé content, décrivant le contenu du fichier à télécharger.

### Advanced settings

**DataStore Path**

Chemin d’accès au fichier Credential contenant le jeton de renouvellement.

**Remarque** : Lorsque l’ID du client, le secret du client ou tout autre paramètre relatif à l’authentification de l’application installée change, vous devez supprimer ce fichier Credential manuellement, avant d’exécuter à nouveau votre Job.

Cette propriété est disponible uniquement lorsque l’option **Installed Application (Id & Secret)** ou **Installed Application (JSON)** est sélectionnée dans la liste déroulante **OAuth Method**.

**Export Google Doc as**

 Sélectionnez le type du document Google Doc à exporter.

**Export Google Draw as**

 Sélectionnez le type du document Google Draw à exporter.

**Export Google Presentation as**

 Sélectionnez le type du document Google Presentation à exporter.

**Export Google Spreadsheet as**

 Sélectionnez le type du document Google Spreadsheet à exporter.

**Add extension**

Cochez cette case pour ajouter une extension au fichier exporté.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

**ERROR_MESSAGE**

Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

**FILE_ID**

ID du fichier. Cette variable est une variable After et retourne une chaîne de caractères.
**Utilisation**

| Règle d'utilisation | Ce composant peut être utilisé en standalone ou en tant que composant de début dans un Job ou un sous-job. |

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tGoogleDriveList

Ce composant liste tous les fichiers ou les dossiers, ou les fichiers et les dossiers d'un dossier Google Drive spécifié.

**Propriétés du tGoogleDriveList Standard**

Ces propriétés sont utilisées pour configurer le tGoogleDriveList s'exécutant dans le framework de Jobs Standard.

Le composant tGoogleDriveList Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant. |

| Application Name | Nom de l’application requis par Google Drive pour obtenir l’accès à ses API. |

<table>
<thead>
<tr>
<th>OAuth Method</th>
<th>Sélectionnez une méthode OAuth utilisée pour accéder à Google Drive dans la liste déroulante.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Access Token (deprecated)</strong> : utilise un jeton d’accès pour accéder à Google Drive.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Installed Application (Id &amp; Secret)</strong> : utilise l’ID et le secret du client créés via la Console Google API pour accéder à Google Drive. Pour plus d’informations concernant cette méthode, consultez Google Identity Platform &gt; Installed applications (en anglais).</td>
</tr>
<tr>
<td></td>
<td>• <strong>Installed Application (JSON)</strong> : utilise un fichier JSON de secret, créé via la Console Google API contenant l’ID du client, son secret et d’autres paramètres OAuth 2.0 pour accéder à Google Drive.</td>
</tr>
</tbody>
</table>
**Service Account** : utilise un fichier JSON de compte de service, créé via la Console Google API, pour accéder à Google Drive. Pour plus d’informations concernant cette méthode, consultez [Google Identity Platform > Service accounts](en anglais).

Pour des informations plus détaillées relatives à l’accès à Google Drive via chaque méthode, consultez [Méthodes OAuth pour accéder à Google Drive](à la page 1342).

| Client ID et Client Secret | ID et secret du client. Ces deux propriétés sont disponibles uniquement lorsque l’option Installed Application (Id & Secret) est sélectionnée dans la liste déroulante OAuth Method. |
| Client Secret JSON | Chemin d’accès au fichier JSON contenant le secret du client. Cette propriété est disponible uniquement lorsque l’option Installed Application (JSON) est sélectionnée dans la liste déroulante OAuth Method. |
| Service Account JSON | Chemin d’accès au fichier JSON du compte de service. Cette propriété est disponible uniquement lorsque l’option Service Account est sélectionnée dans la liste déroulante OAuth Method. |
| Use Proxy | Cochez cette case, lorsque vous travaillez derrière un proxy. Lorsque cette case est cochée, vous devez spécifier la valeur des paramètres suivants :
- **Host** : adresse IP du serveur du proxy HTTP.
- **Port** : numéro du port du serveur du proxy HTTP. |
| Use SSL | Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive. Une fois cette case cochée, vous devez spécifier la valeur des paramètres suivants :
- **Algorithm** : nom de l’algorithme de chiffrement SSL.
- **Keystore File** : chemin d’accès au fichier TrustStore du certificat contenant la liste des certificats auxquels le client fait confiance.
- **Password** : mot de passe utilisé pour vérifier l’intégrité des données TrustStore. |
| Folder Name | Saisissez le nom ou l’ID du dossier duquel les fichiers/dossiers seront listés. |
| Access Method | Sélectionnez la méthode par laquelle le dossier est spécifié, by Name ou by Id. |
### FileList Type

Sélectionnez le type de données à lister.
- **Files**: fichiers uniquement.
- **Directories**: dossiers uniquement.
- **Both**: fichiers et dossiers.

### Include SubDirectories

Cochez cette case pour lister également les fichiers/ dossiers des sous-répertoires.

### Schema et Edit schema

Un schéma est une description de lignes et définit les champs à traiter et à passer au composant suivant.

Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton [...] à côté du champ **Edit schema** pour voir le schéma prédéfini contenant les colonnes suivantes :
- **id**: ID du fichier/dossier.
- **name**: nom du fichier/dossier.
- **mimeType**: type MIME du fichier/dossier.
- **modifiedTime**: date de dernière modification du fichier/dossier.
- **size**: taille du fichier en octets.
- **kind**: type de ressource.
- **trashed**: fichier supprimé ou non.
- **parents**: ID du dossier parent.
- **webViewLink**: lien pour ouvrir le fichier dans un éditeur ou un outil de visualisation Google dans un navigateur.

### Advanced settings

#### DataStore Path

Chemin d’accès au fichier Credential contenant le jeton de renouvellement.

**Remarque** : Lorsque l’ID du client, le secret du client ou tout autre paramètre relatif à l’authentification de l’application installée change, vous devez supprimer ce fichier Credential manuellement, avant d’exécuter à nouveau votre Job.

Cette propriété est disponible uniquement lorsque l’option **Installed Application (Id & Secret)** ou **Installed Application (JSON)** est sélectionnée dans la liste déroulante **OAuth Method**.

#### Include trashed files

Cochez cette case pour prendre en compte également les fichiers et dossiers ayant été supprimés du chemin spécifié.

#### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.
**Variables globales**

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie. |

**Scénario associé**

Gestion des fichiers avec Google Drive à la page 1374
**tGoogleDrivePut**

Ce composant charge des données d'un flux de données ou d'un fichier local, dans Google Drive.

**Propriétés du tGoogleDrivePut Standard**

Ces propriétés sont utilisées pour configurer le tGoogleDrivePut s'exécutant dans le framework de Jobs Standard.

Le composant tGoogleDrivePut Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n'est pas disponible lorsqu'un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant. |

| Application Name | Nom de l'application requis par Google Drive pour obtenir l'accès à ses API. |

<table>
<thead>
<tr>
<th>OAuth Method</th>
<th>Sélectionnez une méthode OAuth utilisée pour accéder à Google Drive dans la liste déroulante.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- <strong>Access Token (deprecated)</strong> : utilise un jeton d'accès pour accéder à Google Drive.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Installed Application (Id &amp; Secret)</strong> : utilise l'ID et le secret du client créés via la Console Google API pour accéder à Google Drive. Pour plus d'informations concernant cette méthode, consultez Google Identity Platform &gt; Installed applications (en anglais).</td>
</tr>
<tr>
<td></td>
<td>- <strong>Installed Application (JSON)</strong> : utilise un fichier JSON de secret, créé via la Console Google API contenant l'ID du client, son secret et d'autres paramètres OAuth 2.0 pour accéder à Google Drive.</td>
</tr>
</tbody>
</table>
- **Service Account**: utilise un fichier JSON de compte de service, créé via la Console Google API, pour accéder à Google Drive. Pour plus d’informations concernant cette méthode, consultez Google Identity Platform > Service accounts (en anglais).

Pour des informations plus détaillées relatives à l’accès à Google Drive via chaque méthode, consultez Méthodes OAuth pour accéder à Google Drive à la page 1342.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Client ID et Client Secret</td>
<td>ID et secret du client. Ces deux propriétés sont disponibles uniquement lorsque l’option Installed Application (Id &amp; Secret) est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td>Client Secret JSON</td>
<td>Chemin d’accès au fichier JSON contenant le secret du client. Cette propriété est disponible uniquement lorsque l’option Installed Application (JSON) est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td>Service Account JSON</td>
<td>Chemin d’accès au fichier JSON du compte de service. Cette propriété est disponible uniquement lorsque l’option Service Account est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
</tbody>
</table>
| Use Proxy | Cochez cette case, lorsque vous travaillez derrière un proxy. Lorsque cette case est cochée, vous devez spécifier la valeur des paramètres suivants :
- **Host**: adresse IP du serveur du proxy HTTP.
- **Port**: numéro du port du serveur du proxy HTTP. |
| Use SSL | Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive. Une fois cette case cochée, vous devez spécifier la valeur des paramètres suivants :
- **Algorithm**: nom de l’algorithme de chiffrement SSL.
- **Keystore File**: chemin d’accès au fichier TrustStore du certificat contenant la liste des certificats auxquels le client fait confiance.
- **Password**: mot de passe utilisé pour vérifier l’intégrité des données TrustStore. |
| File Title | Saisissez le nom du fichier après chargement. |
| Destination Folder | Saisissez le nom ou l’ID du dossier dans lequel les données chargées seront stockées. |
### Access Method

Sélectionnez la méthode par laquelle est spécifié le dossier de destination, by Name ou by Id.

### Replace if Existing

Cochez cette case pour écraser tout fichier existant par le fichier nouvellement chargé.

### Upload Mode

Sélectionnez dans la liste déroulante l’un des modes de chargement suivants :

- **Upload Incoming content as File** : sélectionnez cette option pour charger des données depuis un flux d’entrée provenant du composant précédent.
- **Upload Local File** : sélectionnez cette option pour charger des données depuis un fichier local. Dans le champ File qui s’affiche, spécifiez le chemin d’accès au fichier à charger.
- **Expose As OutputStream** : sélectionnez cette option pour exposer le flux de sortie de ce composant, pouvant être utilisé par d’autres composants, afin d’écrire le contenu du fichier. Par exemple, vous pouvez utiliser la fonctionnalité *Use Output Stream* du composant `tFileOutputDelimited` pour alimenter un flux de sortie exposé du `tGoogleDrivePut`. Pour plus d’informations, consultez `tFileOutputDelimited` à la page 1169.

### Schema et Edit schema

Un schéma est une description de lignes et définit les champs à traiter et à passer au composant suivant.

Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton [...] à côté de *Edit schema* afin de voir le schéma prédéfini contenant les colonnes suivantes :

- **content** : contenu des données chargées.
- **parentFolderId** : ID du dossier parent.
- **fileId** : ID du fichier.

### Advanced settings

#### DataStore Path

Chemin d’accès au fichier Credential contenant le jeton de renouvellement.

**Remarque** : Lorsque l’ID du client, le secret du client ou tout autre paramètre relatif à l’authentification de l’application installée change, vous devez supprimer ce fichier Credential manuellement, avant d’exécuter à nouveau votre Job.

Cette propriété est disponible uniquement lorsque l’option Installed Application (Id & Secret) ou Installed Application (JSON) est sélectionnée dans la liste déroulante OAuth Method.
**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaine de caractères.</td>
</tr>
<tr>
<td><strong>PARENT_FOLDER_ID</strong></td>
<td>ID du dossier parent. Cette variable est une variable After et retourne une chaine de caractères.</td>
</tr>
<tr>
<td><strong>FILE_ID</strong></td>
<td>ID du fichier. Cette variable est une variable After et retourne une chaine de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

**Règle d’utilisation**

Ce composant peut être utilisé en standalone, pour charger un fichier local dans Google Drive ou encore comme composant de sortie pour charger des données dans Google Drive provenant du flux d’entrée d’un composant précédent.

### Gestion des fichiers avec Google Drive

Ce scénario décrit un Job chargeant deux fichiers dans un dossier vide Talend dans le répertoire racine de Google Drive, créant un nouveau dossier Talend Backup dans le répertoire racine, copiant un des deux fichiers dans le nouveau dossier Talend Backup et affichant tous les fichiers et dossiers dans le répertoire racine de Google Drive, dans la console.
Créer un Job pour gérer des fichiers avec Google Drive

Procédure
1. Créez un nouveau Job et ajoutez un composant **tGoogleDriveConnection**, deux **tGoogleDrivePut**, un **tFileInputRaw**, un **tGoogleDriveCreate**, un **tGoogleDriveCopy**, un **tGoogleDriveList** et cinq **tLogRow** dans le Job.
2. Reliez le premier `tGoogleDrivePut` au premier `tLogRow` à l'aide d'un lien `Row > Main`.

3. Répétez l'opération pour relier le `tFileInputRaw` au second `tGoogleDrivePut`, le second `tGoogleDrivePut` au deuxième `tLogRow`, le `tGoogleDriveCreate` au troisième `tLogRow`, le `tGoogleDriveCopy` au quatrième `tLogRow`, et le `tGoogleDriveList` au cinquième `tLogRow`.

4. Reliez le `tGoogleDriveConnection` au premier `tGoogleDrivePut` à l'aide d'un lien `Trigger > On Subjob Ok`.

5. Répétez l'opération pour relier le premier `tGoogleDrivePut` au `tFileInputRaw`, le `tFileInputRaw` au `tGoogleDriveCreate`, le `tGoogleDriveCreate` au `tGoogleDriveCopy` et le `tGoogleDriveCopy` au `tGoogleDriveList`.

**Ouvrir une connexion à Google Drive**

Configurez le `tGoogleDriveConnection` pour vous connecter à Google Drive à l'aide d'un fichier JSON contenant le secret du client.
Avant de commencer

- Le fichier JSON contenant le secret du client doit avoir été téléchargé dans un dossier local via la Console Google API. Pour plus d’informations, consultez Accéder à Google Drive à l’aide d’un fichier JSON contenant le secret de client à la page 1346.

- Un dossier vide Talend doit avoir été créé dans le répertoire racine de Google Drive.

Procédure

1. Double-cliquez sur le tGoogleDriveConnection pour ouvrir sa vue Basic settings.

2. Dans le champ Application Name, saisissez le nom de l’application requis par Google Drive pour obtenir l’accès à son API. Dans cet exemple, saisissez TalendProject.

3. Sélectionnez Installed Application (JSON) dans la liste OAuth Method.

4. Dans le champ Client Secret JSON, spécifiez le chemin d’accès au fichier JSON généré contenant le secret du client, D:/client_secret.json dans cet exemple.

Charger des fichiers dans Google Drive

Procédure

1. Double-cliquez sur le premier tGoogleDrivePut pour ouvrir sa vue Basic settings.
2. Sélectionnez le composant qui créera la connexion à Google Drive, dans la liste déroulante Connection Component, tGoogleDriveConnection_1 dans cet exemple.

3. Sélectionnez by Name dans la liste déroulante Access Method et, dans le champ Destination Folder, saisissez le nom du dossier dans lequel le fichier sera chargé, Talend dans cet exemple.

   Remarque : Lorsque vous accédez à une ressource Google Drive par son nom, si le nom correspond à plus d'une ressource, une erreur est retournée, car la ressource ne peut être identifiée précisément. Dans ce cas, vous pouvez spécifier la ressource Google Drive à l'aide d'un pseudo-chemin hiérarchique, comme/Talend/Documentation. Cet exemple spécifie un dossier nommé Documentation dans le dossier Talend, dans le dossier root de Google Drive.

4. Dans le champ File Name, saisissez le nom du fichier après chargement. Dans cet exemple, saisissez Talend Customers.csv.

5. Sélectionnez Upload Local File dans la liste déroulante Upload Mode et, dans le champ File, parcourez votre système et saisissez le chemin d'accès au fichier à charger. Dans cet exemple, le chemin est D:/Downloads/Talend Customers.csv.

6. Double-cliquez sur le tFileInputRaw. Dans sa vue Basic settings, sélectionnez Read the file as a bytes array dans la zone Mode et spécifiez le chemin d'accès au fichier dont le contenu sera chargé, dans le champ Filename, D:/Downloads/Talend Products.txt dans cet exemple.

7. Double-cliquez sur le second tGoogleDrivePut pour ouvrir sa vue Basic settings.

8. Répétez les étapes 2 à la page 1378 et 3 à la page 1378 afin de configurer ce composant.


10. Sélectionnez Upload Incoming content as File dans la liste Upload Mode.
**Créer un nouveau dossier dans Google Drive**

**Procédure**

1. Double-cliquez sur le `tGoogleDriveCreate` pour ouvrir sa vue `Basic settings`.

   ![Basic settings](image)

2. Sélectionnez dans la liste déroulante **Connection Component** le composant qui créera la connexion à Google Drive, `tGoogleDriveConnection_1` dans cet exemple.
5. Double-cliquez sur le troisième `tLogRow` pour ouvrir sa vue `Basic settings`.
6. Dans la zone **Mode**, sélectionnez **Vertical (each row is a key/value list)** pour un affichage optimal des résultats.

**Copier un fichier dans le nouveau dossier**

**Procédure**

1. Double-cliquez sur le `tGoogleDriveCopy` pour ouvrir sa vue `Basic settings`.

   ![Basic settings](image)

2. Sélectionnez le composant qui créera la connexion à Google Drive, dans la liste **Connection Component**, `tGoogleDriveConnection_1` dans cet exemple.
3. Sélectionnez **File** dans la liste déroulante **Copy Mode**.
5. Dans le champ **Destination Folder Name**, saisissez le nom du dossier dans lequel copier le fichier. Dans cet exemple, saisissez `Talend Backup`. 
6. Cochez la case **Rename (set new title)** et, dans le champ **Destination Name**, saisissez un nouveau nom pour le fichier après copie dans le dossier de destination. Dans cet exemple, saisissez **Talend Customers v1.0.csv**.

7. Double-cliquez sur le quatrième **tLogRow** pour ouvrir sa vue **Basic settings**.

8. Dans la zone **Mode**, sélectionnez **Vertical (each row is a key/value list)** pour un affichage optimal des résultats.

**Lister les fichiers et dossiers dans Google Drive**

**Procédure**

1. Double-cliquez sur le composant **tGoogleDriveList** pour ouvrir sa vue **Basic settings**.

2. Sélectionnez le composant qui créera la connexion à Google Drive, dans la liste déroulante **Connection Component**, le **tGoogleDriveConnection_1** dans cet exemple.

3. Dans le champ **Folder Name**, saisissez le nom du dossier duquel les fichiers et dossiers seront listés. Dans cet exemple, le répertoire est le répertoire racine de Google Drive et vous pouvez utiliser l’alias **root** pour y faire référence.

4. Sélectionnez **Both** dans la liste déroulante **FileList Type**, afin de lister les fichiers et les dossiers du répertoire racine.

5. Cochez la case **Include SubDirectories** pour lister également les fichiers et dossiers des sous-répertoires.

6. Double-cliquez sur le cinquième composant **tLogRow** pour ouvrir sa vue **Basic settings**.

7. Dans la zone **Mode**, sélectionnez **Vertical (each row is a key/value list)** pour un affichage optimal des résultats.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.

2. Exécutez le Job en appuyant sur **F6** ou en cliquant sur l’onglet **Run** dans l’onglet **Run**.
<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>content</td>
<td>ID:CustomerName</td>
</tr>
<tr>
<td>parentFolderId</td>
<td><code>DEE2ZtkTEMASALpaXZf3QpMDDG4xw0O</code></td>
</tr>
<tr>
<td>fileId</td>
<td><code>lfsf/VT0U_RW0ID89q0j_65vi6aMjWvWg</code></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>content</td>
<td>ID:Product</td>
</tr>
<tr>
<td>parentFolderId</td>
<td><code>DEE2ZtkTEMASALpaXZf3QpMDDG4xw0O</code></td>
</tr>
<tr>
<td>fileId</td>
<td><code>1PkhDgKfN2Z80x15_vHaFzSKoonsfM1SFR1</code></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>parentFolderId</td>
<td><code>DEE2ZtkTEMASALpaXZf3QpMDDG4xw0O</code></td>
</tr>
<tr>
<td>fileId</td>
<td><code>lfsf/VT0U_RW0ID89q0j_65vi6aMjWvWg</code></td>
</tr>
<tr>
<td>modifyId</td>
<td><code>1D4LqT-xqhlfjrrIllRnnVBRMNZET6eless</code></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td><code>ltsd-7aXroedvNv5FefeR0o9-1qVt_B1</code></td>
</tr>
<tr>
<td>name</td>
<td>Talend Open Studio for Data Integration</td>
</tr>
<tr>
<td>minaTypa</td>
<td>application/vnd.google-apps.folder</td>
</tr>
<tr>
<td>modifiedTime</td>
<td>2016-07-27T20:59:14Z</td>
</tr>
<tr>
<td>size</td>
<td>null</td>
</tr>
<tr>
<td>kind</td>
<td>drive#file</td>
</tr>
<tr>
<td>trashed</td>
<td>false</td>
</tr>
<tr>
<td>parents</td>
<td><code>0A0Sxt4BHSALp6y09VA</code></td>
</tr>
</tbody>
</table>
Comme affiché dans la capture d'écran ci-dessus, deux fichiers Talend Products.txt et Talend Customers.csv ont été chargés dans le dossier Talend. Un nouveau dossier Talend Backup a été créé dans le dossier racine et le fichier Talend Customers.csv a été copié dans le nouveau dossier et renommé Talend Customers v1.0.csv. Finalement, les fichiers et dossiers du répertoire racine sont listés et affichés dans la console.
tGPGDecrypt

Ce composant appelle la commande `gpg -d` afin de décrypter un fichier crypté GnuPG et sauvegarde le fichier décrypté dans le répertoire spécifié.

**Propriétés du tGPGDecrypt Standard**

Ces propriétés sont utilisées pour configurer le tGPGDecrypt s'exécutant dans le framework de Jobs Standard.

Le composant tGPGDecrypt Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Input encrypted file</th>
<th>Chemin d'accès au fichier crypté file.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output decrypted file</td>
<td>Chemin d'accès au fichier décripté.</td>
</tr>
<tr>
<td>GPG binary path</td>
<td>Chemin d'accès à la commande GPG.</td>
</tr>
<tr>
<td>Passphrase</td>
<td>Saisissez la phrase secrète utilisée pour crypter le fichier d'entrée spécifié. Pour saisir la phrase secrète, cliquez sur le bouton [...] à côté du champ Passphrase, puis, dans la boîte de dialogue qui s'ouvre, saisissez la phrase secrète entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>No TTY Terminal</td>
<td>Cochez cette case pour spécifier qu'aucun terminal TTY n'est utilisé en ajoutant l'option --no-tty à la commande de décryptage.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | FILE : nom du fichier de sortie. Cette variable est une variable Flow et retourne une chaîne de caractères. FILEPATH : chemin d'accès au fichier de sortie. Cette variable est une variable Flow et retourne une chaîne de caractères. ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |

|
Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d'utilisation | Ce composant peut être utilisé en standalone. |

### Scénario : Décrypter un fichier crypté GnuPG et afficher son contenu

Le scénario suivant décrit un Job de trois composants qui décrypte un fichier crypté GnuPG et affiche son contenu dans la console de la vue **Run**.

![Diagramme de scénario](image)

### Déposer et relier les composants

**Procédure**

1. Déposez un composant **tGPGDecrypt**, un **tFileInputDelimited** et un **tLogRow** de la **Palette** dans l'espace de modélisation graphique.
2. Reliez le composant **tGPGDecrypt** au **tFileInputDelimited** à l'aide d'un lien **Trigger > OnSubjobOk** puis connectez le **tFileInputDelimited** au **tLogRow** en utilisant un lien **Row > Main**.

### Configurer les composants

**Procédure**

1. Double-cliquez sur le composant **tGPGDecrypt** afin d'ouvrir sa vue **Component** et définir ses propriétés :
2. Dans le champ **Input encrypted file**, parcourez votre répertoire jusqu’au fichier à décrypter.

3. Dans le champ **Output decrypted file**, saisissez le chemin d’accès au fichier décrypté.

   **Avertissement :**
   Si le chemin d’accès du fichier contient des caractères accentués, vous obtiendrez un message d’erreur lors de l’exécution du Job.

4. Dans le champ **GPG binary path**, parcourez votre répertoire jusqu’au fichier de commandes GPG.


6. Double-cliquez sur le composant **tFileInputDelimited** afin d’ouvrir sa vue **Component** et définir ses propriétés :

   7. Dans la liste **Property Type**, sélectionnez **Built-In**.

   8. Dans le champ **File name/Stream**, définissez le chemin d’accès au fichier décrypté, le chemin de sortie défini dans votre composant **tGPGDecrypt**.


   10. Dans le champ **Footer**, saisissez le nombre de lignes à ignorer à la fin du fichier. Dans cet exemple, aucune ligne n’est à ignorer, saisissez donc 0.


   12. Dans la liste **Schema**, sélectionnez **Built-In**. Cela signifie que les propriétés que vous configurez dans le composant ne seront pas réutilisables dans d’autres Jobs.

   13. Cliquez sur **Edit schema** et éditez le schéma du composant. Cliquez deux fois sur le bouton [+ ] pour ajouter deux colonnes que vous nommerez **idState** et **labelState**.

   14. Cliquez sur **OK** afin de valider vos modifications et fermer l’éditeur.
15. Double-cliquez sur le composant tLogRow afin d’afficher sa vue Component et définir ses propriétés.

17. Dans la zone Mode, sélectionnez l’option Table (print values in cells of a table).

Sauvegarder et exécuter le Job

Procédure
1. Appuyez sur Ctrl+S pour sauvegarder votre Job.
2. Appuyez sur F6 ou cliquez sur le bouton Run de l’onglet Run pour l’exécuter.
**Résultats**

Le fichier spécifié est décrypté et le nombre défini de lignes s'affiche dans la console de la vue Run.
tGreenplumBulkExec

Ce composant améliore les performances pendant les opérations d’Insert dans une base de données Greenplum.

Les composants tGreenplumOutputBulk et tGreenplumBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tGreenplumOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Le tGreenplumBulkExec effectue une action d’Insert sur les données fournies.

Propriétés du tGreenplumBulkExec Standard

Ces propriétés sont utilisées pour configurer le tGreenplumBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tGreenplumBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Remarque</strong> : Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
<td></td>
</tr>
</tbody>
</table>

1388
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit exister pour que l’opération d’Insert soit autorisée.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée : <strong>None</strong> : n’effectuer aucune opération de table. <strong>Drop and create the table</strong> : supprimer la table puis en créer une nouvelle. <strong>Create a table</strong> : créer une table qui n’existe pas encore. <strong>Create table if doesn’t exist</strong> : créer la table si nécessaire. <strong>Clear a table</strong> : supprimer le contenu de la table.</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom du fichier à charger.</td>
</tr>
</tbody>
</table>

**Avertissement :**

Le fichier est situé sur la machine spécifiée par l’URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
</tbody>
</table>
**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Advanced settings

| Action on data | Sélectionnez l’opération que vous voulez effectuer :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bulk insert</strong> ou <strong>Bulk update</strong>. Les informations demandées seront différentes en fonction de l’action choisie.</td>
<td></td>
</tr>
<tr>
<td><strong>Copy the OID for each row</strong></td>
<td>Récupère les identifiants d’objet pour chaque ligne.</td>
</tr>
<tr>
<td><strong>Contains a header line with the names of each column in the file</strong></td>
<td>Spécifiez que la table contient des lignes d’en-tête.</td>
</tr>
<tr>
<td><strong>File type</strong></td>
<td>Sélectionnez le type de fichiers à traiter.</td>
</tr>
<tr>
<td><strong>Null string</strong></td>
<td>Chaîne de caractères affichée pour signifier que la valeur est nulle.</td>
</tr>
<tr>
<td><strong>Fields terminated by</strong></td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td><strong>Escaped char</strong></td>
<td>Caractère d’échappement de la ligne.</td>
</tr>
</tbody>
</table>
Text enclosure | Caractères utilisés pour entourer le texte.
--- | ---
**Force not null for columns** | Définissez la nullabilité des colonnes.
*Force not null* : Cochez la case correspondant à la colonne *(Column)* que vous souhaitez définir comme n’étant pas nulle.

**tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant.

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec un composant tGreenplumOutputBulk. Ensemble, ils offrent un gain de performance important pour l’alimentation d’une base de données Greenplum.
--- | ---
**Dynamic settings** | Cliquez sur le bouton *[+] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table *Dynamic settings* est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue *Basic settings*. Lorsqu’un paramètre dynamique est configuré, la liste *Component List* de la vue *Basic settings* devient inutilisable.


### Scénarios associés

Pour plus d’informations relatives au fonctionnement de tGreenplumBulkExec, consultez les scénarios suivants, dans :

* Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du composant tMysqlOutputBulk.
• Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du composant tMysqlOutputBulkExec.

• Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du composant tOracleBulkExec.
tGreenplumClose

Ce composant ferme une connexion à la base de données Greenplum.

Propriétés du tGreenplumClose Standard

Ces propriétés sont utilisées pour configurer le tGreenplumClose s'exécutant dans le framework de Jobs Standard.

Le composant tGreenplumClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tGreenplumConnection dans la liste.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. |
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez [Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte](#) à la page 2641 et [Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement](#) à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le [Guide utilisateur du Studio Talend](#).

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
Ce composant commité une transaction globale en une fois au lieu de commiter chaque ligne ou chaque lot de lignes et permet donc un gain de performance.

Le tGreenplumCommit valide les données traitées dans un Job à partir d’une base de données connectée. Ce composant utilise une connexion unique.

**Propriétés du tGreenplumCommit Standard**

Ces propriétés sont utilisées pour configurer le tGreenplumCommit s’exécutant dans le framework de Jobs Standard.

Le composant tGreenplumCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant <strong>tGreenplumConnection</strong> dans la liste.</td>
</tr>
</tbody>
</table>
| Close connection | Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.  

**Avertissement :**

Si vous utilisez un lien de type **Row > Main** pour relier le **tGreenplumCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants Greenplum et notamment avec les composants <code>tGreenplumConnection</code> et <code>tGreenplumRoll back</code>.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  
  
La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.  
  

### Scénario associé

Pour consulter des scénarios associés au composant `tGreenplumCommit`, consultez :

- Scénario : Mapper les données à l’aide d’une jointure implicite simple à la page 724.
- Scénario : Insérer des données dans des tables mère/fille à la page 2620.
**tGreenplumConnection**

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Le composant tGreenplumConnection ouvre une connexion vers une base de données afin d’effectuer une transaction.

**Propriétés du tGreenplumConnection Standard**

Ces propriétés sont utilisées pour configurer le tGreenplumConnection s’exécutant dans le framework de Jobs Standard.

Le composant tGreenplumConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
### Use or register a shared DB Connection

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un composant **tRunJob** ayant une de ces options activée fera échouer votre Job.

### Advanced settings

#### Auto Commit

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commit pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

#### tStatCatcher Statistics

Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé avec des composants Greenplum, notamment les composants **tGreenplumCommit** et **tGreenplumRollback**.

### Scénarios associés

Pour un scénario associé au composant **tGreenplumConnection**, consultez :

- **Scénario : Mapper les données à l’aide d’une jointure implicite simple** à la page 724.
• `tMysqlConnection` à la page 2618.
tGreenplumGPLoad

Ce composant permet de charger en masse des données dans une table Greenplum soit à partir d’un fichier de données existant, soit à partir d’un flux de données en mode streaming, soit à partir d’un tube nommé.

Le tGreenplumGPLoad insère des données dans la table d’une base de données Greenplum via l’utilitaire gpload de Greenplum.

Propriétés du tGreenplumGPLoad Standard

Ces propriétés sont utilisées pour configurer le tGreenplumGPLoad s’exécutant dans le framework de Jobs Standard.

Le composant tGreenplumGPLoad Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Nom exact du schéma.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Nom de la table dans laquelle les données doivent être insérées.</td>
</tr>
</tbody>
</table>
**Create table if not exists** : créer la table si nécessaire.

**Drop and create the table** : supprimer la table puis en créer une nouvelle.

**Drop table if exists and create** : supprimer la table si elle existe déjà et la (re)créer.

**Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.

**Action on data**

Vous pouvez effectuer différentes actions sur les données de la table définie :

**Insert** : Ajouter de nouvelles entrées à la table. Si des doublons sont trouvés, le Job s'arrête.

**Update** : Apporter des modifications aux entrées existantes.

**Merge** : Mettre à jour ou ajouter des données dans la table.

**Avertissement :**

Il est nécessaire de spécifier au moins une colonne comme étant une clé primaire sur laquelle baser les opérations **Update** et **Merge**. Vous pouvez le faire en cliquant sur Edit Schema et en cochant la (les) case(s) à côté de la (des) colonne(s) que vous souhaitez définir comme clé primaire(s). Pour configurer les options des opérations **Update** et **Merge**, cochez les cases de la colonne **Match Column** correspondant aux noms des colonnes que vous souhaitez utiliser comme base pour les opérations **Update** et **Merge**. Ensuite, cochez les cases de la colonne **Update Column** correspondant aux noms des colonnes que vous souhaitez mettre à jour. Pour configurer le champ **Update condition**, saisissez la condition qui sera utilisée pour mettre à jour les données.

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

**Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Data file

Chemin d'accès complet au fichier de données à utiliser. Si ce composant est utilisé seul (non connecté à un autre composant par un flux d'entrée), alors saisissez le nom d'un fichier existant à charger dans la base de données. S'il est connecté à un autre composant par un flux d'entrée, saisissez le nom du fichier à générer et à écrire avec les données d'entrée, afin de l'utiliser plus tard avec gpload pour charger dans la base de données. Ce champ est masqué lorsque la case **Use named-pipe** est cochée.

### Use named-pipe

Cochez cette case afin d'utiliser un tube nommé à la place d'un fichier de données. Cette option ne peut être utilisée que lorsque le composant est connecté à un autre composant par un flux d'entrée. Quand la case est cochée, aucun fichier de données n'est généré, et les données sont transférées à l'utilitaire gpload via un tube nommé. Cette option améliore grandement les performances sous Linux et Windows.

**Remarque** :

Ce composant, en mode tube nommé, utilise une interface JNI pour créer un tube nommé et écrire dedans, dans toute plateforme Windows. Le chemin d'accès associé au fichier DLL JNI doit être configuré dans le chemin d'accès à la bibliothèque Java. Le composant est automatiquement fourni dans le Studio Talend avec deux DLL pour les systèmes d'exploitation de 32 et 64 bits.

### Named-pipe name

Spécifiez un nom pour le tube nommé à utiliser. Vérifiez que le nom saisi est valide.

### Die on error

Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

- **Use existing control file (YAML formatted)** : Cochez cette case pour fournir un fichier de contrôle à utiliser avec l'utilitaire gpload au lieu de spécifier
<p>| <strong>Control file</strong> | Saisissez le chemin d’accès au fichier de contrôle à utiliser, entre guillemets doubles, ou cliquez sur le bouton [...] afin de parcourir votre répertoire jusqu’au fichier de contrôle. Cette option est passée à l’utilitaire gpload via l’argument (-f). |
| <strong>CSV mode</strong> | Cochez cette case pour inclure des paramètres spécifiques au format CSV, tels que <strong>Escape char</strong> et <strong>Text enclosure</strong>. |
| <strong>Field separator</strong> | Caractère, chaîne de caractères ou expression régulière utilisé pour séparer les champs. |
| | <strong>Avertissement</strong> : |
| | Argument delim de l’utilitaire gpload. La valeur par défaut est (|). Pour améliorer les performances, utilisez la valeur par défaut. |
| <strong>Escaped char</strong> | Caractère d’échappement de la ligne. |
| <strong>Text enclosure</strong> | Caractères utilisés pour entourer le texte. |
| <strong>Header (skips the first row of data file)</strong> | Cochez cette case pour ignorer la première ligne du fichier de données. |
| <strong>Additional options</strong> | Définissez les arguments gpload dans le tableau correspondant. Cliquez sur le bouton ([+)] autant de fois que nécessaire afin d’ajouter les arguments au tableau. |
| | Cliquez sur le champ <strong>Parameter</strong> et choisissez l’argument souhaité dans la liste, puis cliquez sur le champ <strong>Value</strong> correspondant et saisissez une valeur entre guillemets. |
| <strong>LOCAL_HOSTNAME</strong> | Nom de l’hôte ou adresse IP de la machine sur laquelle gpload fonctionne. Si cette machine est configurée avec plusieurs cartes réseau, vous pouvez spécifier le nom de l’hôte ou l’adresse IP de chaque carte réseau, afin de permettre au trafic d’utiliser simultanément toutes les cartes réseau. Par défaut, l’hôte local ou adresse IP principal(e) est utilisé(e). |
| <strong>PORT (gpfdist port)</strong> | Numéro du port spécifique devant être utilisé par le programme de distribution du fichier gpfdist. Vous pouvez également renseigner <strong>PORT_RANGE</strong> afin de sélectionner un port disponible dans l’écart spécifié. Si <strong>PORT</strong> et <strong>PORT_RANGE</strong> sont spécifiés, <strong>PORT</strong> prévaut. Si aucun des deux n’est défini, un port disponible entre 8000 et 9000 est sélectionné par défaut. Si plusieurs noms d’hôtes sont déclarés dans <strong>LOCAL_HOSTNAME</strong>, le numéro du port est utilisé pour tous les hôtes. Cette configuration est souhaitée si vous... |</p>
<table>
<thead>
<tr>
<th><strong>voulez utiliser toutes les cartes réseau pour charger le même fichier ou le même ensemble de fichiers dans un répertoire donné.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PORT_RANGE</strong> : Peut être utilisé à la place de PORT (gpfdist port) afin de spécifier une plage de numéros de ports parmi lesquels gpload peut choisir un port disponible pour cette instance du programme de distribution du fichier gpfdist.</td>
</tr>
<tr>
<td><strong>NULL_AS</strong> : Chaîne de caractères représentant une valeur null. La valeur par défaut est \N en mode TEXT et une valeur vide sans guillemets en mode CSV. Tout élément de données source correspondant à cette chaîne de caractères sera considéré comme valeur null.</td>
</tr>
<tr>
<td><strong>FORCE_NOT_NULL</strong> : En mode CSV, traite chaque colonne spécifiée comme si elle était entourée de guillemets et n’était donc pas une valeur NULL. Pour la chaîne de caractères null par défaut en mode CSV (aucun caractère entre deux séparateurs), cela crée des valeurs manquantes à évaluer comme des chaines de caractères de longueur zéro.</td>
</tr>
<tr>
<td><strong>ERROR_LIMIT (2 or higher)</strong> : Active le mode d’isolation d’une ligne pour cette opération de chargement. Lorsqu’il est activé et que le nombre maximal d’erreurs n’est pas atteint pour les instances de segments Greenplum durant le traitement de l’entrée, toutes les lignes correctes sont chargées et les lignes ayant des erreurs de format sont rejetées ou enregistrées dans la table <strong>ERROR_TABLE</strong>, si elle est disponible. Si le nombre limite d’erreurs est atteint, les lignes d’entrée ayant des erreurs de format arrêtent l’opération de chargement. Notez que l’isolation d’une ligne en erreur s’applique uniquement aux lignes de données ayant des erreurs de format, par exemple, des attributs manquants ou supplémentaires, des attributs d’un mauvais type de données ou des séquences d’encodage client invalides. Des erreurs de contraintes, telles que les violations de clé primaire, provoquent toujours l’arrêt du chargement. Lorsque cette option n’est pas activée, l’opération de chargement s’arrête à la première erreur rencontrée.</td>
</tr>
<tr>
<td><strong>ERROR_TABLE</strong> : Lorsque la limite <strong>ERROR_LIMIT</strong> est déclarée, spécifie une table d’erreur dans laquelle les lignes contenant des erreurs de format sont enregistrées lors de l’exécution en mode isolation d’une ligne. Vous pouvez observer cette table d’erreurs afin de voir les lignes en erreur n’ayant pas été chargées (s’il y en a).</td>
</tr>
<tr>
<td><strong>Log file</strong></td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
</tr>
</tbody>
</table>
Specify gpload path

Cochez cette case afin de spécifier le chemin d'accès complet à l'exécutable gpload. Vous devez cocher cette case si le chemin d'accès n'est pas spécifié dans la variable d'environnement PATH.

Full path to gpload executable

Chemin d'accès complet à l'exécutable gpload sur la machine en cours d'utilisation. Il est recommandé de spécifier le chemin d'accès dans la variable d'environnement PATH au lieu de sélectionner cette option.

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GPLOAD_OUTPUT : les informations de sortie lors de l'exécution de l'utilitaire gpload. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est principalement utilisé lorsque des transformations particulières ne sont pas requises sur les données à charger dans la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce composant peut être utilisé en standalone ou en tant que composant de sortie.</td>
</tr>
</tbody>
</table>

| Limitation | Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton Install dans l'onglet Component. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet Modules de la perspective Integration de votre studio. Vous pouvez trouver plus d'informations concernant l'installation |
Scénario associé

Pour un scénario associé, consultez Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du composant tMysqlOutputBulkExec.
**tGreenplumInput**

Ce composant lit une base de données et en extrait des champs à l’aide de requêtes.

Le tGreenplumInput exécute une requête de base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

**Propriétés du tGreenplumInput Standard**

Ces propriétés sont utilisées pour configurer le tGreenplumInput s’exécutant dans le framework de Jobs Standard.

Le composant tGreenplumInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><img src="image" alt="Configuration de connexion" /></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue <strong>Basic settings</strong> du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case lorsque vous utilisez le composant tGreenplumConnection.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Nom exact du schéma</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| **Query type et Query** | Saisissez votre requête de base de données en respectant l’ordre des champs défini dans le schéma. |
| **Guess Query** | Cliquez sur le bouton **Guess Query** pour générer la requête qui correspond au schéma de votre table, dans le champ **Query**. |
| **Guess schema** | Cliquez sur le bouton **Guess schema** pour récupérer le schéma de la table. |
Advanced settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use cursor</td>
<td>Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d'optimiser les performances.</td>
</tr>
<tr>
<td>Trim all the String/Char columns</td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td>Trim column</td>
<td>Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>QUERY</td>
<td>requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Greenplum.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple...</td>
<td></td>
</tr>
</tbody>
</table>
ple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénarios associés

Pour des scénarios associés, consultez :

- Scénario : Mapper les données à l’aide d’une jointure implicite simple à la page 724.

Consultez également Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520.
tGreenplumOutput

Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.
Le tGreenplumOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

Propriétés du tGreenplumOutput Standard

Ces propriétés sont utilisées pour configurer le tGreenplumOutput s’exécutant dans le framework de Jobs Standard.
Le composant tGreenplumOutput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>![Icone d'assistant de configuration]</td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>![Remarque]</td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par</td>
</tr>
</tbody>
</table>
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :

  * None : n’effectuer aucune opération de table.
  * Drop and create the table : supprimer la table puis en créer une nouvelle.
  * Create a table : créer une table qui n’existe pas encore.
  * Create table if doesn’t exist : créer la table si nécessaire.
  * Drop a table if exists and create : supprimer la table si elle existe déjà, puis en créer une nouvelle.
  * Clear a table : supprimer le contenu de la table. |
| Action on data | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

  * Insert : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
  * Update : Mettre à jour les entrées existantes.
  * Insert or update : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. |
**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

**Delete** : Supprimer les entrées correspondantes au flux d’entrée.

⚠ **Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

---

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects.

### Advanced settings

| **Commit every** | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution. |
| **Additional Columns** | Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier. |

**Name** : Saisissez le nom de la colonne à modifier ou à insérer.

**SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.

**Position** : Sélectionnez Before, Replace ou After, en fonction de l’action à effectuer sur la colonne de référence.

**Reference column** : Saisissez une colonne de référence que le composant tPostgresqlOutput peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

**Use field options** : Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.

**Use Batch** : Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.

**Remarque** :

Cette case est disponible lorsque vous sélectionnez Insert, Update, ou Delete dans la liste Action on data.
| Batch Size | Spécifiez le nombre d’enregistrements à traiter dans chaque lot.  
Ce champ est disponible uniquement lorsque la case Use batch mode est cochée. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| Global Variables | NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.  
NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.  
NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.  
NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.  
NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.  
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.  
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données Greenplum. Il permet aussi de créer un flux de rejet avec un lien Row &gt; Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de</td>
</tr>
</tbody>
</table>
contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Scénarios associés

Pour un scénario associé au composant **tGreenplumOutput** consultez :

- **Scénario : Mapper les données à l’aide d’une jointure implicite simple** à la page 724.
- **Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput** à la page 2667 du **tMysqlOutput**.
tGreenplumOutputBulk

Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Greenplum.

Les composants tGreenplumOutputBulk et tGreenplumBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tGreenplumOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Écrit un fichier composé de colonnes et basé sur le séparateur défini et sur les standards Greenplum.

Propriétés du tGreenplumOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tGreenplumOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tGreenplumOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Either Built-in or Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom du fichier à générer.</td>
</tr>
<tr>
<td>Avertissement</td>
<td>Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.</td>
</tr>
<tr>
<td>Append</td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
</tbody>
</table>
### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.


  Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

  Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Row separator</strong></th>
<th>Chaîne (ex : <code>\n</code> sous Unix) séparant les lignes.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field separator</strong></td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td><strong>Include header</strong></td>
<td>Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.</td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez <em>Custom</em> et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</td>
</tr>
</tbody>
</table>
**tStatCatcher Statistics**

| Cochez cette case pour collecter les données de log au niveau du composant. |

---

**Global Variables**


---

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec le composant tGreenplumBulkExec. Ensemble, ils offrent un gain de performance important pour l’alimentation d’une base de données Greenplum. |

---

**Scénarios associés**

Pour un scénario associé au tGreenplumOutputBulk, consultez :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du composant tMysqlOutputBulk.
- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec.
tGreenplumOutputBulkExec

Ce composant est un composant dédié qui permet un gain de performance pendant les opérations d’Insert dans une base de données Greenplum.

Les composants tGreenplumOutputBulk et tGreenplumBulkExec sont généralement utilisés ensemble comme deux parties d’un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l’opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tGreenplumOutputBulkExec.

Le tGreenplumOutputBulkExec effectue une action d’Insert sur les données fournies.

Propriétés du tGreenplumOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tGreenplumOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tGreenplumOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de la base de données. Seuls localhost, 127.0.0.1 ou l’adresse IP exacte de la machine locale permettent un fonctionnement optimal. Le serveur de la base de données doit être installé sur la même machine que le Studio Talend ou que le Job contenant un tGreenplumOutputBulkExec.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Nom du schéma.</td>
</tr>
</tbody>
</table>
| **Username et Password** | Informations d'authentification de l'utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Table** | Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit déjà exister pour que l'opération d'insert soit autorisée. |
| **Action on table** | Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :  
**None** : n'effectuer aucune opération de table.  
**Drop and create the table** : supprimer la table puis en créer une nouvelle.  
**Create a table** : créer une table qui n'existe pas encore.  
**Create table if doesn't exist** : créer la table si nécessaire.  
**Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer un nouvelle.  
**Clear a table** : supprimer le contenu de la table. |
| **File Name** | Nom du fichier à générer et à charger.  
**Avertissement** :  
Ce fichier est généré sur la machine spécifiée par l'URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.  
**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.  
Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com). |
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Advanced settings

<table>
<thead>
<tr>
<th>Action on data</th>
<th>Sélectionnez l'opération que vous voulez effectuer : <strong>Bulk insert</strong> ou <strong>Bulk update</strong>. Les informations demandées seront différentes en fonction de l'action choisie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy the OID for each row</td>
<td>Récupère les identifiants d'objet pour chaque ligne.</td>
</tr>
<tr>
<td>Contains a header line with the names of each column in the file</td>
<td>Spécifiez que la table contient des lignes d’en-tête.</td>
</tr>
<tr>
<td>File type</td>
<td>Sélectionnez le type de fichiers à traiter.</td>
</tr>
<tr>
<td>Null string</td>
<td>Chaîne de caractères affichée pour signifier que la valeur est nulle.</td>
</tr>
<tr>
<td>Fields terminated by</td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td>Escaped char</td>
<td>Caractère d'échappement de la ligne.</td>
</tr>
<tr>
<td>Text enclosure</td>
<td>Caractères utilisés pour entourer le texte.</td>
</tr>
<tr>
<td>Force not null for columns</td>
<td>Définissez la nullabilité des colonnes <strong>Force not null</strong> : Cochez la case correspondant à la colonne (<strong>Column</strong>) que vous souhaitez définir comme n'étant pas nulle.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Le serveur de la base de données doit être installé sur la même machine que le Studio Talend ou que le Job contenant un tGreenplumOutputBulkExec, afin que le composant fonctionne correctement.</td>
</tr>
</tbody>
</table>

Scénarios associés

Pour un scénario associé au tGreenplumOutputBulkExec, consultez :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du composant tMysqlOutputBulk.
- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec.
**tGreenplumRollback**

Ce composant évite le commit de transaction involontaire.
Le tGreenplumRollback annule la transaction dans une base de données connectée.

**Propriétés du tGreenplumRollback Standard**

Ces propriétés sont utilisées pour configurer le tGreenplumRollback s'exécutant dans le framework de Jobs Standard.
Le composant tGreenplumRollback Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant de connexion <strong>tGreenplumConnection</strong> dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en association avec des composants Greenplum, notamment avec le **tGreenplumConnection** et le **tGreenplumCommit**. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant... |
la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Pour un scénario associé au tGreenplumRollback, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623 du composant tMysqlRollback.
tGreenplumRow

Selon la nature de la requête et de la base de données, ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler).

Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Le tGreenplumRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tGreenplumRow Standard

Ces propriétés sont utilisées pour configurer le tGreenplumRow s'exécutant dans le framework de Jobs Standard.

Le composant tGreenplumRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
la vue **Basic settings** du composant de connexion créant cette connexion.

2. **Au niveau enfant,** utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend.*

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma</td>
</tr>
</tbody>
</table>

**Username et Password**

Informations d’authentification de l’utilisateur de base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password,** puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

- **Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend.*

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository,** trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la
Table Name: Nom de la table à lire.
Query type: Peut être Built-in ou Repository.

**Built-in:** Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.

**Repository:** Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.

Guess Query: Cliquez sur le bouton Guess Query pour générer la requête correspondant au schéma de votre table dans le champ Query.

Query: Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.

Die on error: Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects.

### Advanced settings

**Propagate QUERY’s recordset**: Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column.

**Remarque** :
Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type Object. Ce composant est généralement suivi du tParseRecordSet.


**Parameter Index** : Saisissez la position du paramètre dans l’instruction SQL.

**Parameter Type** : Saisissez le type du paramètre.

**Parameter Value** : Saisissez la valeur du paramètre.

**Remarque** :
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

**Commit every**

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.

ERROR _MESSAGE_ : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.
Scénarios associés

Pour un scénario associé, consultez :

- **Scénario : Combiner deux flux pour une sortie sélective** à la page 2706.
- **Procédure du composant tDBSQLRow**.
- **Scénario : Supprimer et re-générer un index de table MySQL** à la page 2700 du composant tMysqlRow.
tGreenplumSCD

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.
Le tGreenplumSCD reflète et traque les modifications d’une table Greenplum SCD dédiée.

**Propriétés du tGreenplumSCD Standard**

Ces propriétés sont utilisées pour configurer le tGreenplumSCD s’exécutant dans le framework de Jobs Standard.
Le composant tGreenplumSCD Standard appartient aux familles Business Intelligence et Databases.
Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remarque :</strong></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td>1.</td>
<td>Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue <strong>Basic settings</strong> du composant de connexion créant cette connexion.</td>
</tr>
</tbody>
</table>
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma de la base de données.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur de la base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Table        | Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réserve line lors du nommage des champs.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
- View schema : sélectionnez cette option afin de voir le schéma.  
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].  

**SCD Editor**
L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension.
Pour plus d’informations, consultez Méthodologie de gestion du SCD à la page 2716.

**Use memory saving Mode**
Cochez cette case pour améliorer les performances du système.

**Source keys include Null**
Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null.

⚠️ Avertissement : Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.

**Die on error**
Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

**Advanced settings**

**End date time details**
Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format **HH:mm:ss**. La valeur par défaut pour ce champ est **12:00:00**.
Ce champ apparaît uniquement lorsqu’un SCD de **Type 2** est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD.

**Debug mode**
Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

**Variables globales**

**Global Variables**

**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier.

**NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier.

**NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton <strong>[+]</strong> pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Limitation

Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD.

### Scénario associé

Pour un scénario associé, consultez tMysqlSCD à la page 2712.
**tGroovy**

Le tGroovy permet d’étendre les fonctionnalités du Job *Talend* grâce au langage Groovy, qui est une syntaxe simplifiée du Java.

Le tGroovy permet de saisir du code personnalisé afin de l’intégrer dans le programme *Talend*. Ce code est exécuté une seule fois.

**Propriétés du tGroovy Standard**

Ces propriétés sont utilisées pour configurer le tGroovy s’exécutant dans le framework de Jobs Standard.

Le composant tGroovy Standard appartient à la famille Custom Code.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Groovy Script</th>
<th>Saisissez le code Groovy que vous souhaitez exécuter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables</td>
<td>Ce tableau comprend deux colonnes :</td>
</tr>
<tr>
<td></td>
<td><em>Name</em> : nom de la variable appelée dans le code.</td>
</tr>
<tr>
<td></td>
<td><em>Value</em> : valeur associée à cette variable.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option. Une variable <em>Flow</em> fonctionne durant l’exécution d’un composant. Une variable <em>After</em> fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <em>Ctrl+Espace</em> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le <em>Guide utilisateur du Studio Talend</em>.</th>
</tr>
</thead>
</table>
**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en stand-alone, ou en tant que sous-job à un seul composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Il est nécessaire de connaître le langage Groovy.</td>
</tr>
</tbody>
</table>

**Scénarios associés**

- Pour un scénario utilisant du Code Groovy, consultez Scénario : Appeler du code Groovy contenu dans un fichier à la page 1438.
- Pour un exemple d’utilisation proche dans le mode de fonctionnement, consultez Scénario : Imprimer le contenu d’une variable à la page 1940 du composant tJava.
**tGroovyFile**

Ce composant permet d’étendre les fonctionnalités du Job Talend grâce au langage Groovy, qui est une syntaxe simplifiée du Java.

Le tGroovyFile permet d’appeler un script existant développé en Groovy.

**Propriétés du tGroovyFile Standard**

Ces propriétés sont utilisées pour configurer le tGroovyFile s’exécutant dans le framework de Jobs Standard.

Le composant tGroovyFile Standard appartient à la famille Custom Code.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Groovy File</th>
<th>Nom et chemin d’accès au fichier contenant votre code Groovy.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Variables</th>
<th>Ce tableau comprend deux colonnes :</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>nom de la variable appelée dans le code.</td>
</tr>
<tr>
<td>Value</td>
<td>valeur associée à cette variable.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

**Global Variables**

|-------------------|--------------------------------------------------------------------------|

Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant peut être utilisé en standalone, ou en tant que sous-job à un seul composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Il est nécessaire de connaître le langage Groovy.</td>
</tr>
</tbody>
</table>

**Scénario : Appeler du code Groovy contenu dans un fichier**

Ce scénario utilise un composant *tGroovyFile* en standalone. Le Job appelle un fichier contenant du code Groovy afin d’afficher les informations de ce fichier dans la **Console**.

![tGroovyFile_1](image)

**Construire le Job**

Développez le dossier **Custom Code** de la Palette et déposez un composant *tGroovyFile* dans l’espace de modélisation graphique.

**Configurer le composant tGroovyFile**

**Procédure**

1. Double-cliquez sur le composant pour ouvrir sa vue **Component**.

2. Dans le champ **Groovy File**, saisissez le chemin d’accès au fichier contenant le code Groovy, ou parcourrez votre système jusqu’à ce fichier. Dans cet exemple, le chemin est : "D:/Input/Agedu capitaine.txt". Le fichier contient le code Groovy suivant :

   ```groovy
   println("The captain is " + age + " years old")
   ```

3. Dans la table **Variables**, ajoutez une ligne en cliquant sur le bouton [+].

4. Dans la colonne **Name**, saisissez "age" puis, dans la colonne **Value**, saisissez 50.
Exécuter le Job

Procédure

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour exécuter le Job.

La **Console** affiche les informations contenues dans le fichier d’entrée, auxquelles est ajouté le résultat de la variable.

```plaintext
Starting job tGroovyFile at 09:49 17/02/2010.
[statistics] connecting to socket on port 4016
[statistics] connected
The captain is 50 years old
[statistics] disconnected
Job tGroovyFile ended at 09:49 17/02/2010. [exit code=0]
```
tGSBucketCreate

Ce composant permet de créer un nouveau bucket que vous pouvez utiliser afin d’organiser des données et contrôler les accès aux données dans Google Cloud Storage.

### Propriétés du tGSBucketCreate Standard

Ces propriétés sont utilisées pour configurer le tGSBucketCreate s’exécutant dans le framework de Jobs Standard.

Le composant tGSBucketCreate Standard appartient aux familles Big Data et Cloud.

Le composant de ce framework est toujours disponible.

#### Basic settings

| **Use an existing connection** | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component list pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |
| **Access Key et Secret Key** | Saisissez les informations d’authentification obtenues de Google afin d’effectuer des requêtes sur Google Cloud Storage.  
Ces clés peuvent être consultées dans la vue Interoperable Access, sous l’onglet Google Cloud Storage du projet dans la console d’API de Google.  
Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.  
Les champs Access Key et Secret Key ne sont disponibles que si la case Use an existing connection n’est pas cochée. |
| **Bucket name** | Spécifiez le nom du bucket que vous souhaitez créer.  
Notez que le nom du bucket doit être unique au sein du système Google Cloud Storage.  
Pour plus d’informations sur les conventions de nommage des buckets, consultez https://developers.google.com/storage/docs/bucketnaming (en anglais). |
| **Special configure** | Cochez cette case afin de fournir une configuration supplémentaire au bucket à créer. |
| **Project ID** | Spécifiez l’identifiant du projet auquel le nouveau bucket appartient. |
**Location**

Sélectionnez dans la liste l’emplacement du nouveau bucket. Actuellement, Europe et US sont disponibles. Par défaut, l’emplacement du bucket est US.

Notez qu’une fois le bucket créé à un emplacement spécifique, il ne peut être déplacé vers un nouvel emplacement.

**Acl**

Dans la liste, sélectionnez la liste de contrôle d’accès (ACL) souhaitée pour le nouveau bucket.

Selon l’ACL du bucket, les accès requis par les utilisateurs sont acceptés ou rejetés. Si vous ne spécifiez pas d’ACL prédéfinie pour le nouveau bucket, l’ACL privée prédéfinie du projet s’applique.


**Die on error**

Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

---

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

---

**Global Variables**

**Global Variables**

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

---

**Utilisation**

**Règle d’utilisation**

Ce composant est généralement utilisé avec un tGSBucketList afin de vérifier si le nouveau bucket est correctement créé.
Scénario associé

Pour des scénarios associés, consultez Scénario : Vérifier l’absence d’un bucket, le créer et lister tous les buckets S3 à la page 3521.
tGSBucketDelete

Ce composant supprime un bucket vide dans Google Cloud Storage afin de libérer des ressources occupées.

Propriétés du tGSBucketDelete Standard

Ces propriétés sont utilisées pour configurer le tGSBucketDelete s’exécutant dans le framework de Jobs Standard.

Le composant tGSBucketDelete Standard appartient aux familles Big Data et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component list pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access Key et Secret Key</strong></td>
<td>Saisissez les informations d’authentification obtenues de Google afin d’effectuer des requêtes sur Google Cloud Storage.</td>
</tr>
<tr>
<td></td>
<td>Ces clés peuvent être consultées dans la vue Interoperable Access, sous l’onglet Google Cloud Storage du projet dans la console d’API de Google.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td></td>
<td>Les champs Access Key et Secret Key ne sont disponibles que si la case Use an existing connection n’est pas cochée.</td>
</tr>
<tr>
<td><strong>Bucket name</strong></td>
<td>Spécifiez le nom du bucket que vous souhaitez supprimer. Assurez-vous que le bucket à supprimer est vide.</td>
</tr>
<tr>
<td></td>
<td><strong>Avertissement :</strong></td>
</tr>
<tr>
<td></td>
<td>La suppression du bucket ne peut être annulée. Vous devez donc vous assurer d’avoir sauvegardé toutes les données que vous souhaitez garder avant de continuer.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.</td>
</tr>
</tbody>
</table>
Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec un tGSBucketList afin de vérifier si les données sont bien importées. |

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tGSBucketExist**

Ce composant vérifie l’existence d’un bucket dans Google Cloud Storage afin d’effectuer plus d’opérations.

**Propriétés du tGSBucketExist Standard**

Ces propriétés sont utilisées pour configurer le tGSBucketExist s’exécutant dans le framework de Jobs Standard.

Le composant tGSBucketExist Standard appartient aux familles Big Data et Cloud.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component list pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
</table>
| **Access Key et Secret Key** | Saisissez les informations d’authentification obtenues de Google afin d’effectuer des requêtes sur Google Cloud Storage.  
Ces clés peuvent être consultées dans la vue Interoperable Access, sous l’onglet Google Cloud Storage du projet dans la console d’API de Google.  
Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.  
Les champs Access Key et Secret Key ne sont disponibles que si la case Use an existing connection n’est pas cochée. |
| **Bucket name** | Spécifiez le nom du bucket dont vous souhaitez vérifier l’existence dans Google Cloud Storage. |
| **Die on error** | Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur. |

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |
Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>BUCKET_EXIST : présence ou non d'un bucket spécifié. Cette variable est une variable Flow et retourne un booléen.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BUCKET_NAME : nom d'un bucket spécifié. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Utilisation

| Règle d'utilisation | Ce composant est utilisable en tant que composant standalone. |

Scénario associé

Pour des scénarios associés, Scénario : Vérifier l’absence d’un bucket, le créer et lister tous les buckets S3 à la page 3521.
tGSBucketList

Ce composant récupère une liste des buckets de tous les projets ou d'un projet spécifique dans Google Cloud Storage.

Le tGSBucketList effectue une boucle sur tous les buckets de tous les projets ou d'un projet spécifique dans Google Cloud Storage.

Propriétés du tGSBucketList Standard

Ces propriétés sont utilisées pour configurer le tGSBucketList s'exécutant dans le framework de Jobs Standard.

Le composant tGSBucketList Standard appartient aux familles Big Data et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component list pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify project ID</td>
<td>Cochez cette case et, dans le champ Project ID, spécifiez l’identifiant du projet dont vous voulez récupérer la liste des buckets.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics   | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |
Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Définition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURRENT_BUCKET_NAME</td>
<td>nom du bucket courant. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>NB_BUCKET</td>
<td>nombre de buckets. Cette variable est une variable After et retourne un nombre entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Définition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce composant est utilisable en tant que composant standalone ou en tant que composant de début d’un Job.</td>
</tr>
</tbody>
</table>

Scénario associé

Pour un scénario associé, consultez **Scénario : Vérifier l’absence d’un bucket, le créer et lister tous les buckets S3** à la page 3521.
**tGSClose**

Ce composant ferme une connexion active sur Google Cloud Storage afin de libérer les ressources occupées.

**Propriétés du tGSClose Standard**

Ces propriétés sont utilisées pour configurer le tGSClose s’exécutant dans le framework de Jobs Standard.

Le composant tGSClose Standard appartient aux familles Big Data et Cloud.

Le composant de ce framework est toujours disponible.

### Basic settings

| Component List | Dans la liste, sélectionnez le composant tGSConnection si plus d’une connexion est prévue pour le Job courant. |

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Global Variables


### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Google Cloud Storage, en particulier le tGSConnection. |
Scénario associé

Pour un scénario dans lequel le tGSClose est utilisé, consultez Scénario : Gérer des fichiers avec Google Cloud Storage à la page 1465.
tGSConnection

Ce composant fournit les informations d'authentification nécessaires pour effectuer des requêtes sur le système Google Cloud Storage et permet de réutiliser la connexion créée pour Google Cloud Storage.

Propriétés du tGSConnection Standard

Ces propriétés sont utilisées pour configurer le tGSConnection s’exécutant dans le framework de Jobs Standard.

Le composant tGSConnection Standard appartient aux familles Big Data et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

| Access Key et Secret Key | Saisissez les informations d'authentification obtenues de Google afin d'effectuer des requêtes sur Google Cloud Storage. 
Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google.
Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace |
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec d'autres composants Google Cloud Storage, en particulier le tGSClose. |

**Scénario associé**

Pour un scénario dans lequel le tGSConnection est utilisé, consultez Scénario : Gérer des fichiers avec Google Cloud Storage à la page 1465.
Ce composant copie ou déplace des objets au sein d’un bucket ou entre plusieurs buckets dans Google Cloud Storage. Le tGSCopy permet de rationaliser les traitements en effectuant des tâches de copie de manière automatique.

**Propriétés du tGSCopy Standard**

Ces propriétés sont utilisées pour configurer le tGSCopy s’exécutant dans le framework de Jobs Standard. Le composant tGSCopy Standard appartient aux familles Big Data et Cloud. Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Source bucket name</td>
<td>Saisissez le nom du bucket à partir duquel vous souhaitez copier ou déplacer des objets.</td>
</tr>
<tr>
<td>Source object key</td>
<td>Saisissez la clé de l’objet à copier.</td>
</tr>
<tr>
<td>Source is folder</td>
<td>Cochez cette case si l’objet source est un dossier.</td>
</tr>
<tr>
<td>Target bucket name</td>
<td>Saisissez le nom du bucket vers lequel vous souhaitez copier ou déplacer des objets.</td>
</tr>
</tbody>
</table>
**Target folder**
Saisissez le dossier cible vers lequel les objets sont copiés ou déplacés.

**Action**
Sélectionnez l’action que vous souhaitez effectuer sur les objets de la liste.
- **Copy** : copie les objets depuis le bucket ou le dossier source vers le bucket ou dossier cible.
- **Move** : déplace les objets depuis le bucket ou dossier source vers le bucket ou dossier cible.

**Rename**
Cochez cette case et, dans le champ **New name**, saisissez un nouveau nom pour l’objet à copier ou à déplacer.
La case **Rename** n’est pas disponible si vous cochez la case **Source is folder**.

**Die on error**
Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

**Advanced settings**

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

**Global Variables**

**Global Variables**

**SOURCE_BUCKET** : nom du bucket source. Cette variable est une variable **After** et retourne une chaîne de caractères.

**SOURCE_OBJECTKEY** : clé d’un objet source. Cette variable est une variable **After** et retourne une chaîne de caractères.

**DESTINATION_BUCKETNAME** : nom du bucket cible. Cette variable est une variable **After** et retourne une chaîne de caractères.

**DESTINATION_FOLDER** : dossier cible. Cette variable est une variable **After** et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d'utilisation | Ce composant peut être utilisé en standalone. |

**Scénario associé**

Pour un scénario dans lequel le tGSCopy est utilisé, consultez Scénario : Gérer des fichiers avec Google Cloud Storage à la page 1465.
tGSDelete

Ce composant supprime les objets qui correspondent aux critères définis dans Google Cloud Storage afin de libérer des ressources occupées.

Propriétés du tGSDelete Standard

Ces propriétés sont utilisées pour configurer le tGSDelete s’exécutant dans le framework de Jobs Standard.

Le composant tGSDelete Standard appartient aux familles Big Data et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Key et Secret Key</td>
<td>Saisissez les informations d’authentification obtenues de Google afin d’effectuer des requêtes sur Google Cloud Storage.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td></td>
<td>Ces clés peuvent être consultées dans la vue Interoperable Access, sous l’onglet Google Cloud Storage du projet dans la console d’API de Google.</td>
</tr>
<tr>
<td></td>
<td>Les champs Access Key et Secret Key ne sont disponibles que si la case Use an existing connection n’est pas cochée.</td>
</tr>
<tr>
<td>Key prefix</td>
<td>Saisissez le préfixe de la clé des objets à supprimer. Ainsi, seuls les objets dont la clé commence par ce préfixe sont listés.</td>
</tr>
<tr>
<td>Delimiter</td>
<td>Saisissez le séparateur afin de supprimer uniquement les objets avec des noms jusqu’à ce séparateur.</td>
</tr>
<tr>
<td>Specify project ID</td>
<td>Cochez cette case et, dans le champ Project ID, saisissez l’identification du projet dont vous voulez supprimer des objets.</td>
</tr>
<tr>
<td>Delete object from bucket list</td>
<td>Cochez cette case et renseignez la table Bucket pour supprimer des objets dans les buckets spécifiés.</td>
</tr>
</tbody>
</table>
### tGSDelete

- **Bucket name**: saisissez le nom du bucket dans lequel vous souhaitez supprimer les objets.
- **Key prefix**: saisissez le préfixe pour supprimer les objets dont les clés commencent par la chaîne définie dans le bucket spécifié.
- **Delimiter**: saisissez le séparateur afin de lister uniquement les objets avec des noms jusqu’à ce séparateur.

Si vous cochez la case **Delete object from bucket list**, les champs **Key prefix** et **Delimiter**, ainsi que la case **Specify project ID** ne sont pas disponibles.

#### Die on error
Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

#### Advanced settings

- **tStatCatcher Statistics**
  Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

#### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>Nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

#### Utilisation

- **Règle d’utilisation**
  Ce composant peut être utilisé avec le composant **tGSList** afin de vérifier si les objets correspondant aux critères définis sont correctement supprimés.
Scénario associé

Pour un scénario dans lequel le `tGSDelte` est utilisé, consultez Scénario : Gérer des fichiers avec Google Cloud Storage à la page 1465.
tGSGet

Ce composant récupère les objets qui correspondent aux critères définis dans Google Cloud Storage et les envoie vers un dossier local.

Propriétés du tGSGet Standard

Ces propriétés sont utilisées pour configurer le tGSGet s’exécutant dans le framework de Jobs Standard.

Le composant tGSGet Standard appartient aux familles Big Data et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant adéquat dans la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
</table>
| Access Key et Secret Key  | Saisissez les informations d’authentification obtenues de Google afin d’effectuer des requêtes sur Google Cloud Storage.  
Saisissez les informations d’authentification obtenues de Google afin d’effectuer des requêtes sur Google Cloud Storage.  
Ces clés peuvent être consultées dans la vue Interoperable Access, sous l’onglet Google Cloud Storage du projet dans la console d’API de Google.  
Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.  
Les champs Access Key et Secret Key ne sont disponibles que si la case Use an existing connection n’est pas cochée. |
| Key prefix                | Saisissez le préfixe afin de télécharger uniquement les objets dont les clés commencent par le préfixe défini. |
| Delimiter                 | Saisissez le séparateur afin de télécharger uniquement les objets avec des noms jusqu’à ce séparateur. |
| Specify project ID        | Cochez cette case et, dans le champ Project ID, saisissez l’identification du projet dont vous voulez obtenir des objets. |
| Use keys                  | Cochez cette case et renseignez la table Keys afin de définir les critères pour les objets à télécharger depuis Google Cloud Storage. |
- **Bucket name**: saisissez le nom du bucket à partir duquel vous souhaitez télécharger des objets.
- **Key**: saisissez la clé de l’objet à télécharger.
- **New name**: saisissez un nouveau nom pour l’objet à télécharger.

Si vous cochez la case **Use keys**, les champs **Key prefix** et **Delimiter**, ainsi que les cases **Specify project ID** et **Get files from bucket list** ne sont pas disponibles.

<table>
<thead>
<tr>
<th>Get files from bucket list</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case et renseignez la table <strong>Bucket</strong> afin de définir les critères pour les objets à télécharger depuis Google Cloud Storage.</td>
</tr>
<tr>
<td>• <strong>Bucket name</strong>: saisissez le nom du bucket à partir duquel vous souhaitez télécharger des objets.</td>
</tr>
<tr>
<td>• <strong>Key prefix</strong>: saisissez le préfixe pour télécharger des objets dont les clés commencent par le préfixe spécifié dans le bucket défini.</td>
</tr>
<tr>
<td>• <strong>Delimiter</strong>: saisissez le séparateur afin de télécharger les objets avec des noms jusqu’à ce séparateur depuis le bucket défini.</td>
</tr>
</tbody>
</table>

Si vous cochez la case **Get files from bucket list**, les champs **Key prefix** et **Delimiter**, ainsi que les cases **Specify project ID** et **Use keys** ne sont pas disponibles.

<table>
<thead>
<tr>
<th>Output directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spécifiez le dossier dans lequel vous souhaitez stocker les objets téléchargés.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Die on error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong> : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette
Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d'utilisation | Ce composant est généralement utilisé avec d'autres composants Google Cloud Storage, par exemple le tGSPut. |

**Scénario associé**

Aucun scénario n'est disponible pour la version Standard de ce composant.
tGSLList

Ce composant récupère, un à un, une liste d’objets depuis Google Cloud Storage.
Le tGSLList fait une boucle sur une liste d’objets qui correspondent aux critères spécifiés dans Google Cloud Storage.

**Propriétés du tGSLList Standard**

Ces propriétés sont utilisées pour configurer le tGSLList s’exécutant dans le framework de Jobs Standard.
Le composant tGSLList Standard appartient aux familles Big Data et Cloud.
Le composant de ce framework est toujours disponible.

**Basic settings**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant adéquat dans la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>
| **Access Key et Secret Key** | Saisissez les informations d’authentification obtenues de Google afin d’effectuer des requêtes sur Google Cloud Storage.  
Ces clés peuvent être consultées dans la vue Interoperable Access, sous l’onglet Google Cloud Storage du projet dans la console d’API de Google.  
Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.  
Les champs Access Key et Secret Key ne sont disponibles que si la case Use an existing connection n’est pas cochée. |
| **Key prefix** | Saisissez le préfixe de la clé des objets à lister. Ainsi, seuls les objets dont la clé commence par ce préfixe sont listés. |
| **Delimiter** | Saisissez le séparateur afin de lister uniquement les objets avec des noms jusqu’à ce séparateur. |
| **Specify project ID** | Cochez cette case et, dans le champ Project ID, saisissez l’identification du projet dont vous voulez récupérer la liste des objets. |
| **List objects in bucket list** | Cochez cette case et renseignez la table Bucket pour récupérer les objets dans les buckets spécifiés. |
### Bucket name
Saisissez le nom du bucket à partir duquel vous souhaitez récupérer les objets.

### Key prefix
Saisissez le préfixe pour lister uniquement les objets dont les clés commencent par la chaîne définie dans le bucket spécifié.

### Delimiter
Saisissez le séparateur afin de lister uniquement les objets avec des noms jusqu'à ce séparateur.

Si vous cochez la case **List objects in bucket list**, les champs **Key prefix** et **Delimiter**, ainsi que la case **Specify project ID** ne sont pas disponibles.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant. |

### Global Variables

| Global Variables | CURRENT_BUCKET : nom du bucket courant. Cette variable est une variable **Flow** et retourne une chaîne de caractères.  
CURRENT_KEY : clé courante. Cette variable est une variable **Flow** et retourne une chaîne de caractères.  
NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.  
Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.  
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Le composant tGSList peut être utilisé en standalone ou en tant que composant de début. |

### Scénario associé

Pour un scénario dans lequel le tGSList est utilisé, consultez [Scénario : Gérer des fichiers avec Google Cloud Storage](#) à la page 1465.
**tGSPut**

Ce composant met un fichier d'un dossier local dans Google Cloud Storage afin que vous puissiez les gérer avec Google Cloud Storage.

**Propriétés du tGSPut Standard**

Ces propriétés sont utilisées pour configurer le tGSPut s'exécutant dans le framework de Jobs Standard.

Le composant tGSPut Standard appartient aux familles Big Data et Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Access Key et Secret Key</td>
<td>Saisissez les informations d'authentification obtenues de Google pour envoyer des requêtes à Google Cloud Storage.</td>
</tr>
<tr>
<td></td>
<td>Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td></td>
<td>Les champs Access Key et Secret Key ne sont disponibles que si la case Use an existing connection n'est pas cochée.</td>
</tr>
<tr>
<td>Bucket name</td>
<td>Saisissez le nom du bucket dans lequel vous souhaitez charger les fichiers.</td>
</tr>
<tr>
<td>Local directory</td>
<td>Saisissez le chemin d'accès complet ou parcourez votre système vers le dossier local contenant les fichiers à charger.</td>
</tr>
<tr>
<td>Google Storage directory</td>
<td>Saisissez le nom du dossier Google Storage dans lequel vous souhaitez charger des fichiers.</td>
</tr>
<tr>
<td>Use files list</td>
<td>Cochez cette case et renseignez la table Files.</td>
</tr>
</tbody>
</table>
### Filemask
Saisissez un nom ou un masque de fichier à l'aide de caractères spéciaux (*) ou d'expressions régulières.

### New name
Saisissez le nom à donner au fichier après le chargement.

#### Die on error
Cette case est décochée par défaut afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant. |

### Global Variables


### Utilisation

| Règle d’utilisation | Ce composant peut être utilisé avec d’autres composants Google Cloud Storage, par exemple le tGSGet. |

### Scénario : Gérer des fichiers avec Google Cloud Storage

Le scénario suivant permet de créer un Job chargeant des fichiers à partir d’un répertoire local dans un bucket dans Google Storage. Ce Job permet ensuite d’effectuer des opérations de copie, de déplacement et de suppression sur ces fichiers, puis de lister et afficher les fichiers dans leurs buckets respectifs dans la console.
Prérequis : Vous devez posséder un compte Google Cloud Storage et avoir créé trois buckets dans le même répertoire Google Cloud Storage. Dans cet exemple, les buckets créés sont **bighouse**, **bed_room** et **study_room**.

Déposer et relier les composants

Pourquoi et quand exécuter cette tâche

Afin de créer ce Job, procédez comme suit :

Procédure

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un **tGSConnection**, un **tGSPut**, deux **tGSCopy**, un **tGSDelete**, un **tGSLList**, un **tIterateToFlow**, un **tLogRow** et un **tGSClose**.
2. Reliez le **tGSCo**nnection au **tGSPut** à l’aide d’un lien **Trigger > On Subjob Ok**.
3. Reliez le **tGSPut** au premier **tGSCopy** à l’aide d’un lien **Trigger > On Subjob Ok**.
4. Reliez ensuite le premier **tGSCopy** au second **tGSCopy**, le **tGSCopy** au **tGSDelete**, le **tGSDelete** au **tGSList** et le **tGSList** au **tGSClose** à l’aide de liens **Trigger > On Subjob Ok**.
5. Reliez le **tGSList** au **tIterateToFlow** à l’aide d’un lien **Row > Iterate**.
6. Reliez le **tIterateToFlow** au **tLogRow** à l’aide d’un lien **Row > Main**.

**Configurer les composants**

**Ouvrir une connexion à Google Cloud Storage**

**Procédure**

1. Double-cliquez sur le **tGSCo**nnection afin d’ouvrir sa vue **Basic settings** dans l’onglet **Component**.

   ![Configuration d’une connexion à Google Cloud Storage](image)

2. Naviguez vers la console d’API de Google dans votre navigateur afin d’accéder au projet Google contenant les services Cloud Storage que vous souhaitez utiliser.
4. Dans la vue **Component** du Studio, collez la clé d’accès et la clé secrète dans les champs correspondants, respectivement **Access Key** et **Secret Key**.

**Charger les fichiers dans Google Cloud Storage**

**Procédure**

1. Double-cliquez sur le **tGSPut** afin d’ouvrir sa vue **Basic settings** dans l’onglet **Component**.

   ![Configuration de la mise en carte de fichiers](image)

2. Cochez la case **Use an existing connection** puis sélectionnez la connexion que vous avez configurée précédemment.
3. Dans le champ **Bucket name**, saisissez le nom du bucket dans lequel vous souhaitez charger les fichiers, *bighouse* dans cet exemple.
4. Dans le champ **Local directory**, parcourez votre système vers le dossier contenant les fichiers à charger, *D:/Input/House* dans cet exemple.
La capture d'écran suivante montre les fichiers contenus dans ce dossier :

5. Laissez les autres paramètres tels qu’ils sont.

Copier tous les fichiers d’un bucket à un autre

Procédure

1. Double-cliquez sur le premier tGSCopy afin d’ouvrir sa vue Basic settings dans l’onglet Component.

2. Cochez la case Use an existing connection puis sélectionnez la connexion que vous avez configurée précédemment.

3. Dans le champ Source bucket name, saisissez le nom du bucket à partir duquel vous souhaitez copier les fichiers, bighouse dans cet exemple.

4. Cochez la case Source is a folder. Tous les fichiers contenus dans le bucket bighouse sont copiés.

5. Dans le champ Target bucket name, saisissez le nom du bucket vers lequel vous souhaitez copier les fichiers, bed_room dans cet exemple.


Déplacer un fichier d’un bucket à un autre et les renommer

Procédure

1. Double-cliquez sur le second tGSCopy afin d’ouvrir sa vue Basic settings dans l’onglet Component.
2. Cochez la case **Use an existing connection** puis sélectionnez la connexion que vous avez configurée précédemment.

3. Dans le champ **Source bucket name**, saisissez le nom du bucket contenant le fichier que vous souhaitez déplacer, *bighouse* dans cet exemple.


5. Dans le champ **Target bucket name**, saisissez le nom du bucket vers lequel vous souhaitez déplacer le fichier, *study_room* dans cet exemple.

6. Sélectionnez **Move** dans la liste **Action**. Le fichier source défini, *computer_01.txt*, est déplacé du bucket *bighouse* vers le bucket *study_room*.


8. Laissez les autres paramètres tels qu’ils sont.

**Supprimer un fichier dans un bucket**

**Procédure**

1. Double-cliquez sur le **tGSDDelete**, afin d’ouvrir sa vue **Basic settings** dans l’onglet **Component**.
2. Cochez la case **Use an existing connection** puis sélectionnez la connexion que vous avez configurée précédemment.

3. Cochez la case **Delete object from bucket list**. Dans la table **Bucket**, saisissez les informations du fichier que vous souhaitez supprimer.

Dans cet exemple, le fichier *computer_03.csv* est supprimé du bucket *bed_room* dont les fichiers sont copiés depuis le bucket *bighouse*.

**Lister les fichiers contenus dans les trois buckets**

**Procédure**

1. Double-cliquez sur le *tGSList* afin d’ouvrir sa vue **Basic settings** dans l’onglet **Component**.

2. Cochez la case **Use an existing connection** puis sélectionnez la connexion que vous avez configurée précédemment.

3. Cochez la case **List objects in bucket list**. Dans la colonne **Bucket name** de la table **Bucket**, saisissez le nom des trois buckets, *bighouse*, *study_room* et *bed_room*.

4. Double-cliquez sur le *tIterateToFlow* afin d’ouvrir sa vue **Basic settings** dans l’onglet **Component**.

5. Cliquez sur le bouton **Edit schema** afin de définir les données qui sont passées au *tLogRow*.

Dans cet exemple, ajoutez deux colonnes, *bucketName* et *key*, de type **Object**.
6. La table **Mapping** est automatiquement renseignée avec les colonnes définies.
   Dans la colonne Value, saisissez `globalMap.get("tGList_2_CURRENT_BUCKET")` dans la ligne bucketName et `globalMap.get("tGList_2_CURRENT_KEY")` dans la ligne key, respectivement. Vous pouvez également appuyer sur **Ctrl + Espace** puis choisir les variables appropriées.

7. Double-cliquez sur le **tLogRow** afin d’ouvrir sa vue **Basic settings** dans l’onglet **Component**.

8. Sélectionnez l’option **Table (print values in cells of a table)** pour un meilleur affichage des résultats.

**Fermer la connexion à Google Cloud Storage**

**Procédure**

1. Double-cliquez sur le **tGSClose** afin d’ouvrir sa vue **Basic settings** dans l’onglet **Component**.
2. Dans la liste **Component List**, sélectionnez la connexion que vous souhaitez fermer.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder le Job.
2. Appuyez sur **F6** ou cliquez sur **Run** dans l’onglet **Run** afin d’exécuter le Job.
Les fichiers contenus dans les trois buckets s’affichent. Comme attendu, les fichiers du bucket *bighouse* sont d’abord copiés vers le bucket *bed_room*. Ensuite, le fichier *computer_01.txt* du bucket *bighouse* est déplacé vers le bucket *study_room* puis renommé *laptop.txt*. Enfin, le fichier *computer_03.csv* est supprimé du bucket *bed_room*. 

<table>
<thead>
<tr>
<th>bucketName</th>
<th>key</th>
</tr>
</thead>
<tbody>
<tr>
<td>bighouse</td>
<td>books.txt</td>
</tr>
<tr>
<td>bighouse</td>
<td>clothes.txt</td>
</tr>
<tr>
<td>bighouse</td>
<td>computer_02.txt</td>
</tr>
<tr>
<td>bighouse</td>
<td>computer_03.csv</td>
</tr>
<tr>
<td>bighouse</td>
<td>model cars.txt</td>
</tr>
<tr>
<td>bighouse</td>
<td>tables.txt</td>
</tr>
<tr>
<td>bighouse</td>
<td>toys.txt</td>
</tr>
<tr>
<td>study_room</td>
<td>laptop.txt</td>
</tr>
<tr>
<td>bed_room</td>
<td>books.txt</td>
</tr>
<tr>
<td>bed_room</td>
<td>clothes.txt</td>
</tr>
<tr>
<td>bed_room</td>
<td>computer_01.txt</td>
</tr>
<tr>
<td>bed_room</td>
<td>computer_02.txt</td>
</tr>
<tr>
<td>bed_room</td>
<td>model cars.txt</td>
</tr>
<tr>
<td>bed_room</td>
<td>tables.txt</td>
</tr>
<tr>
<td>bed_room</td>
<td>toys.txt</td>
</tr>
</tbody>
</table>
**tHashInput**

Ce composant lit des données dans la mémoire cache, écrites par le composant tHashOutput afin de peupler rapidement de données et de faciliter les transactions comprenant un large volume de données.

Les composants de la famille Technical sont normalement masqués par défaut, dans la Palette. Pour savoir comment les afficher dans la Palette, consultez le Guide utilisateur du Studio Talend.

**Propriétés du tHashInput Standard**

Ces propriétés sont utilisées pour configurer le tHashInput s’exécutant dans le framework de Jobs Standard.

Le composant tHashInput Standard appartient à la famille Technical.

Le composant de ce framework est toujours disponible.

**Basic settings**

| **Schema et Edit schema** | Un schéma est une description de ligne, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant (**Repository**). Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

**Link with a tHashOutput** | Cochez cette case pour connecter votre composant à un tHashOutput. Cette case est cochée par défaut. |
**Component list**

Liste déroulante comprenant les composants **tHashOutput** disponibles.

**Clear cache after reading**

Cochez cette case pour vider le cache après lecture des données chargées par un composant **tHashOutput**. Ainsi, les composants **tHashInput** suivants, s’il y en a, ne pourront lire les données du cache chargées par le **tHashOutput**.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

- **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.

   Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

   Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

   Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

**Règle d’utilisation**

Ce composant est utilisé avec le **tHashOutput**. Il lit des données depuis la mémoire cache, écrites par le **tHashOutput**. Ensemble, ces deux composants permettent un accès très rapide aux données afin de faciliter les transactions comprenant un grand nombre de données.

**Scénario 1 : Lire des données directement dans la mémoire cache afin d’y accéder rapidement**

Le Job suivant lit dans la mémoire cache un large volume de données écrites par deux composants **tHashOutput** et envoie le contenu dans un **tFileOutputDelimited**. L’objectif de ce Job est de montrer la vitesse de lecture et d’écriture des données. En pratique, les données ainsi générées peuvent...
être utilisées comme table de référence dans certains cas où un grand volume de données doit être référencé.

**Déposer et relier les composants**

**Procédure**

1. Glissez-déposez les composants suivants de la Palette dans l'espace de modélisation graphique : deux tFixedFlowInput, deux tHashOutput, un tHashInput et un tFileOutputDelimited.
2. Reliez le premier tFixedFlowInput au premier tHashOutput à l'aide d'un lien Row > Main.
3. Reliez le second tFixedFlowInput au second tHashOutput à l'aide d'un lien Row > Main.
4. Connectez le premier sous-job (composant tFixedFlowInput_1) au deuxième sous-job (composant tFixedFlowInput_2) à l'aide d'un lien OnSubjobOk.
5. Reliez le tHashInput au tFileOutputDelimited à l'aide d'un lien Row > Main.
6. Connectez le deuxième sous-job au dernier sous-job à l'aide d'un lien OnSubjobOk.

![Diagramme des composants](image)

**Configurer les composants**

**Configurer les entrées de données et le cache**

**Procédure**

1. Double-cliquez sur le premier tFixedFlowInput pour afficher sa vue Basic settings et configurer ses propriétés de base.
2. Dans la liste Schema, sélectionnez Built-In.

**Remarque :**

Vous pouvez sélectionner Repository dans la liste Schema afin de renseigner automatiquement les champs, si la métadonnée correspondante est stockée dans le Repository. Pour plus d'informations concernant la centralisation des métadonnées, consultez le chapitre relatif aux métadonnées dans le Guide utilisateur du Studio Talend.

3. Cliquez sur le bouton [...] à côté du champ Edit schema afin de définir la structure des données du flux d'entrée. Dans ce cas, le schéma contient deux colonnes : ID et ID_Insurance. Cliquez sur OK pour fermer la boîte de dialogue.

4. Dans le champ Number of rows, saisissez le nombre d’entrées à écrire, ici 50000.

5. Sélectionnez l’option Use Single Table. Dans la table Values, saisissez, dans la colonne Value, une valeur pour chaque colonne, ici 1 pour ID et 3 pour ID_Insurance.

6. Effectuez la même opération pour le second tFixedFlowInput et saisissez 2 pour ID et 4 pour ID_Insurance.

7. Double-cliquez sur le premier tHashOutput afin d’afficher sa vue Basic settings et définir ses propriétés.
8. Dans la liste Schema, sélectionnez Built-In puis cliquez sur le bouton Sync columns afin de récupérer le schéma du composant précédent. Sélectionnez Keep all dans la liste Keys management et laissez la case Append cochée.


**Configurer la récupération des données du cache et la sortie des données.**

**Procédure**

1. Double-cliquez sur le composant tHashInput pour afficher sa vue Basic settings et définir ses propriétés.

2. Dans la liste Schema, sélectionnez Built-In. Cliquez sur le bouton [...] à côté du champ Edit schema pour définir la structure des données, la même que dans le tHashOutput.

3. Sélectionnez tHashOutput_1 dans la liste Component list.

4. Double-cliquez sur le composant tFileOutputDelimited pour afficher sa vue Basic settings et configurer ses propriétés.

5. Dans la liste Property Type, sélectionnez Built-In. Dans le champ File Name, saisissez le chemin d'accès à votre fichier ou cliquez sur le bouton [...] afin de parcourir votre système jusqu'à votre fichier. Ici, le chemin d'accès est le suivant : "E:/Allr70207V5.0/Talend-All-r70207-V5.0.0NB/workspace/out.csv"
6. Cochez la case Include Header puis cliquez sur le bouton Sync columns afin de récupérer le schéma du composant précédent.

**Sauvegarder et exécuter le Job**

**Procédure**
1. Appuyez sur les touches Ctrl+S pour sauvegarder votre Job.
2. Appuyez sur F6, ou cliquez sur la vue Run, puis cliquez sur Run pour exécuter le Job.

**Résultats**

Vous pouvez voir que les nombreuses entrées ont été écrites et lues de manière très rapide.

**Scénario 2 : Vider la mémoire avant d’y charger les données si une boucle existe dans le même sous-job**

Ce scénario montre l'utilisation de l’option Append du composant tHashOutput, qui permet de supprimer des données répétitives ou indésirables s’il y a une boucle dans le sous-job du tHashOutput.

Pour construire le Job, procédez comme suit :

**Déposer et relier les composants**

**Procédure**
2. Reliez le tLoop autFixedFlowInput à l’aide d’un lien Row > Iterate.
3. Connectez le composant tFixedFlowInput au tHashOutput à l’aide d’un lien Row > Main.
4. Reliez le tHashInput au tLogRow à l’aide d’un lien Row > Main.
5. Connectez le tLoop au tHashInput à l’aide d’un lien OnSubjobOk.
Configurer les composants

Configurer les données d'entrée et le cache

Procédure

1. Double-cliquez sur le tLoop pour afficher sa vue Basic settings.

2. Dans la zone Loop Type, sélectionnez For comme type de boucle. Saisissez respectivement 1, 2 et 1 dans les champs From, To et Step. Laissez cochée la case Values are increasing.

3. Double-cliquez sur le composant tFixedFlowInput pour afficher sa vue Basic settings.
Dans la liste **Schema**, sélectionnez **Built-In**.

**Remarque :**
Vous pouvez sélectionner **Repository** dans la liste **Schema** afin de renseigner automatiquement tous les champs si la métadonnée correspondante est stockée dans le **Repository**. Pour plus d’informations concernant la centralisation des métadonnées sous le nœud **Metadata**, consultez le Guide utilisateur du **Studio Talend**.

5. Cliquez sur **Edit schema** pour définir la structure des données du flux d’entrée. Dans ce scénario, le schéma contient une colonne **Name**.

6. Cliquez sur **OK** pour fermer la boîte de dialogue.

7. Renseignez le champ **Number of rows** afin de spécifier les entrées à écrire, par exemple 1.

8. Cochez la case **Use Single Table**. Dans la table **Values**, assignez une valeur au champ **Name**, par exemple **Marx**.

9. Double-cliquez sur le **tHashOutput** pour afficher sa vue **Basic settings**.

10. Dans la liste **Schema**, sélectionnez **Built-In** et cliquez sur **Sync columns** pour récupérer le schéma du composant précédent. Sélectionnez **Keep all** dans la liste **Keys management** et décochez la case **Append**.

**Configurer la récupération des données du cache et la sortie des données**

**Procédure**

1. Double-cliquez sur le **tHashInput** pour afficher sa vue **Basic settings**.
2. Sélectionnez **Built-In** dans la liste **Schema**. Cliquez sur **Edit schema** afin de définir la même structure des données que dans le schéma du **tHashOutput**.

3. Sélectionnez **tHashOutput_2** dans la liste **Component list**.

4. Double-cliquez sur le **tLogRow** pour afficher sa vue **Basic settings**.

5. Sélectionnez **Built-In** dans la liste **Schema** et cliquez sur **Sync columns** pour récupérer le schéma du composant précédent. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)**.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** pour exécuter le Job, ou cliquez sur le bouton **Run**, dans l’onglet **Run**.

   Vous pouvez constater qu’une seule ligne a été écrite alors que deux ont été générées par le **tFixedFlowInput**.

![Diagram](image-url)
tHashOutput

Ce composant charge des données dans la mémoire cache afin de permettre un accès rapide aux données et de faciliter les transactions comportant un large volume de données.

Les composants de la famille Technical sont normalement masqués par défaut, dans la Palette. Pour savoir comment les afficher dans la Palette, consultez le Guide utilisateur du Studio Talend.

**Propriétés du tHashOutput Standard**

Ces propriétés sont utilisées pour configurer le tHashOutput s’exécutant dans le framework de Jobs Standard.

Le composant tHashOutput Standard appartient à la famille Technical.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit schema | Un schéma est une description de ligne, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant (Repository). Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
| | • View schema : sélectionnez cette option afin de voir le schéma.
| | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
| | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. Cliquez sur Sync columns pour récupérer le schéma du composant précédent dans le Job.
| **Built-in** | Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.
| **Repository** | Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.
| Link with a tHashOutput | Cochez cette case pour connecter votre composant à un tHashOutput. |
Remarque :
Si plusieurs composants `tHashOutput` sont ainsi reliés, les données chargées dans le cache par tous ces composants peuvent être lues par un `tHashInput` relié à l’un d’entre eux.

<table>
<thead>
<tr>
<th>Component list</th>
<th>Liste déroulante comprenant les composants <code>tHashOutput</code> disponibles.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data write model</td>
<td>Liste des modes d’écriture de données disponibles.</td>
</tr>
<tr>
<td>Keys management</td>
<td>Liste des modes de gestion de clés disponibles.</td>
</tr>
<tr>
<td>• Keep all</td>
<td>écrit toutes les données reçues dans la mémoire cache.</td>
</tr>
<tr>
<td>• Keep first</td>
<td>écrit uniquement le premier enregistrement dans la mémoire cache si plusieurs enregistrements sont reçus avec la même valeur de clé.</td>
</tr>
<tr>
<td>Append</td>
<td>Cette case est cochée par défaut. Elle sert à écrire des données dans la mémoire, à la suite, au cas où une itération existe dans le même sous-job. Si la case est décochée, le composant <code>tHashOutput</code> va effacer la mémoire avant d’y charger les données.</td>
</tr>
<tr>
<td>Remarque :</td>
<td>Si la case <code>Link with a tHashOutput</code> est cochée, la case <code>Append</code> est masquée mais reste activée.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant écrit des données dans la mémoire cache. Il est étroitement lié au composant tHashInput. Ensemble, ces composants permettent un accès rapide aux données et facilitent les transactions comportant un large volume de données. |

**Scénarios associés**

Pour un scénario associé, consultez :

- **Scénario 1** : Lire des données directement dans la mémoire cache afin d’y accéder rapidement à la page 1474.

- et **Scénario 2** : Vider la mémoire avant d’y charger les données si une boucle existe dans le même sous-job à la page 1478 du composant tHashInput.
tHBaseClose

Ce composant ferme une connexion à HBase établie dans un Job.
Le tHBaseClose ferme une connexion active à une base de données HBase.

Propriétés du tHBaseClose Standard

Ces propriétés sont utilisées pour configurer le tHBaseClose s'exécutant dans le framework de Jobs Standard.
Le composant tHBaseClose Standard appartient aux familles Big Data et Databases.
Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Component list | S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant tHBaseConnection dans la liste. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec des composants HBase, notamment avec le tHBaseConnecti on. |
Avant de commencer, vérifiez que tous les prérequis de l’IP de Loopback (rebouclage) attendus par votre base de données sont respectés.

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib
\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontrez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue Preferences dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Scénario associé

Pour un exemple d’utilisation du tHBaseClose, consultez Scénario : Echanger des données clients avec HBase à la page 1501.
tHBaseConnection

Ce composant établit une connexion HBase à réutiliser dans d'autres composants HBase de votre Job.
Le tHBaseConnection ouvre une connexion vers une base de données HBase.

Propriétés du tHBaseConnection Standard

Ces propriétés sont utilisées pour configurer le tHBaseConnection s’exécutant dans le framework de Jobs Standard.
Le composant tHBaseConnection Standard appartient aux familles Big Data et Databases.
Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Property type | Peut être Built-in ou Repository.  
- Built-in : Propriétés utilisées ponctuellement.  
- Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées. |
| Distribution | Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
- Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).  
- L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.  
1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,  
2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution. |
personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>HBase version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hadoop version of the distribution</td>
<td>Cette liste s’affiche uniquement lorsque vous avez sélectionné Custom dans la liste des distributions à connecter à un cluster n’étant pas officiellement supporté par le Studio. Dans cette situation, sélectionnez la version de Hadoop de ce cluster personnalisé, Hadoop 1 ou Hadoop 2.</td>
</tr>
<tr>
<td>Zookeeper quorum</td>
<td>Saisissez le nom ou l’URL du service Zookeeper utilisé pour coordonner les transactions entre votre Studio et votre base de données. Notez que, lorsque vous configurez Zookeeper, vous pouvez avoir besoin de configurer explicitement la propriété zookeeper.znode.parent pour définir le chemin vers le nœud znode racine contenant tous les znodes créés et utilisés par</td>
</tr>
</tbody>
</table>
votre base de données. Cochez la case la case **Set Zookeeper znode parent** afin de définir cette propriété.

<table>
<thead>
<tr>
<th>Zookeeper client port</th>
<th>Saisissez le numéro du port d’écoute client du service Zookeeper que vous utiliserez.</th>
</tr>
</thead>
</table>
| **Inspect the classpath for configurations** | Cochez cette case pour permettre au composant de vérifier les fichiers de configuration dans le répertoire configuré pour la variable `$HADOOP_CONF_DIR` et de lire directement les paramètres de ces fichiers dans le répertoire. Cette fonctionnalité vous permet de modifier facilement la configuration Hadoop afin que le composant puisse passer d’un environnement à un autre, comme par exemple pour passer d’un environnement test à un environnement production.  

Dans ce cas, les champs ou les options utilisée pour configurer la connexion Hadoop et/ou la sécurité Kerberos sont masqués.  

Si vous souhaitez utiliser certains paramètres comme les paramètres Kerberos mais que ces paramètres ne sont pas inclus dans les fichiers de configuration Hadoop, vous devez créer un fichier appelé `talend-site.xml` et mettre ce fichier dans le répertoire défini `$HADOOP_CONF_DIR`. Le fichier `talend-site.xml` doit se présenter comme suit :

```xml
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>talend.kerberos.authentication</name>
    <value>kinit</value>
    <description> Set the Kerberos authentication method to use. Valid values are: kinit or keytab. </description>
  </property>
  <property>
    <name>talend.kerberos.keytab.principal</name>
    <value>user@BIGDATA.COM</value>
    <description> Set the keytab's principal name. </description>
  </property>
  <property>
    <name>talend.kerberos.keytab.path</name>
    <value>/kdc/user.keytab</value>
    <description> Set the keytab's path. </description>
  </property>
  <property>
    <name>talend.encryption</name>
    <value>none</value>
```
Les paramètres lus depuis ces fichiers de configuration écrasent ceux utilisés par défaut dans le Studio. Lorsqu’un paramètre n’existe pas dans ces fichiers de configuration, le paramètre par défaut est utilisé.

Use kerberos authentication

Si la base de données choisie utilise la sécurité Kerberos, cochez cette case puis saisissez le nom des principaux dans les champs affichés. Vous pouvez trouver cette information dans le fichier `hbase-site.xml` du cluster utilisé.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case **Use a keytab to authenticate**. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur
que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser.

**Advanced settings**

**Properties**

Si vous devez utiliser la configuration personnalisée pour votre HBase, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) vont écraser celles définies précédemment pour HBase.

Par exemple, vous devez définir la valeur de la propriété `dfs.replication` comme étant 1 pour la configuration de HBase. Vous devez ensuite ajouter une ligne à la table à l’aide du bouton [+] et saisis le nom et la valeur de cette propriété dans la ligne.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

**Règle d’utilisation**

Ce composant est généralement utilisé avec des composants HBase, notamment avec le **tHBaseClose**.

**Prérequis**

Avant de commencer, vérifiez que tous les prérequis de l’IP de Loopback (rebouclage) attendus par votre base de données sont respectés.

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le *Studio Talend* et...
d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRPCient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRPCient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

### Scénario associé

Pour un exemple d’utilisation du composant tHBaseConnection, consultez Scénario : Echanger des données clients avec HBase à la page 1501.
tHBaseInput

Ce composant lit des données d'une base de données HBase et extrait les colonnes sélectionnées.

HBase est une base de données distribuée, orientée colonnes, hébergeant des tables volumineuses et peu peuplées, dans des clusters.

Le tHBaseInput extrait des colonnes selon la définition du schéma. Il passe ensuite les colonnes au composant suivant via un lien **Main > Row**.

**Filtres HBase**

Ce tableau présente les filtres HBase disponibles dans le **Studio Talend** ainsi que les paramètres requis par ces filtres.

<table>
<thead>
<tr>
<th>Filter type (type de filtre)</th>
<th>Filter column</th>
<th>Filter family</th>
<th>Filter operation</th>
<th>Filter value</th>
<th>Filter comparator type</th>
<th>Objectif</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Column Value Filter</td>
<td>Oui</td>
<td>Oui</td>
<td>Oui</td>
<td>Oui</td>
<td>Oui</td>
<td>Compare les valeurs d'une colonne donnée par rapport à la valeur définie du paramètre Filter value. Si les conditions de filtre sont respectées, toutes les colonnes de la ligne sont retournées.</td>
</tr>
<tr>
<td>Family filter</td>
<td></td>
<td>Oui</td>
<td>Oui</td>
<td>Oui</td>
<td></td>
<td>Retourne les colonnes de la famille répondant aux conditions de filtre.</td>
</tr>
<tr>
<td>Qualifier filter</td>
<td></td>
<td>Oui</td>
<td>Oui</td>
<td>Oui</td>
<td></td>
<td>Retourne les colonnes dont le nom (qualifier) répond à la condition de filtre.</td>
</tr>
<tr>
<td>Column prefix filter</td>
<td>Oui</td>
<td></td>
<td>Oui</td>
<td></td>
<td></td>
<td>Retourne toutes les colonnes dont le qualifier contient un préfixe défini dans le paramètre Filter column.</td>
</tr>
<tr>
<td>Multiple column prefix filter</td>
<td>Oui (Différents préfixes doivent être séparés par une virgule, par exemple : id,id_1,id_2.)</td>
<td></td>
<td></td>
<td>Oui</td>
<td></td>
<td>Fonctionne de la même manière que le Column prefix filter mais vous permet de spécifier différents préfixes.</td>
</tr>
<tr>
<td>Column range filter</td>
<td>Oui (Les limites de début et de fin de la plage doivent être séparées par une virgule.)</td>
<td></td>
<td></td>
<td>Oui</td>
<td></td>
<td>Permet un scanner l'intérieur d'une ou plusieurs ligne(s) et de retourner toutes les colonnes correspondantes d'une ligne scannée.</td>
</tr>
</tbody>
</table>
### Propriétés du tHBaseInput Standard

Ces propriétés sont utilisées pour configurer le tHBaseInput s'exécutant dans le framework de Jobs Standard.

Le composant tHBaseInput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

#### Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Value filter</th>
<th>Distribution</th>
</tr>
</thead>
</table>
| Peut être **Built-In** ou **Repository**.  
**Built-In** : propriétés utilisées ponctuellement.  
**Repository** : sélectionnez le fichier dans lequel sont stockées les propriétés du composant. | Cliquez sur cette icône pour ouvrir l’assistant de connexion à la base de données et stocker les paramètres de connexion configurés dans la vue Basic settings.  
Pour plus d’informations concernant la configuration et le stockage des paramètres de connexion à la base de données, consultez le Guide utilisateur du Studio Talend. | Sélectionnez dans la liste le cluster que vous utilisez.  
Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez. Parmi ces options, les suivantes nécessitent une configuration spécifique. |

---

<table>
<thead>
<tr>
<th>Filter type (type de filtre)</th>
<th>Filter column</th>
<th>Filter family</th>
<th>Filter operation</th>
<th>Filter value</th>
<th>Filter comparator type</th>
<th>Objectif</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row filter</td>
<td>Oui</td>
<td>Oui</td>
<td>Oui</td>
<td>Filtre les clés de lignes et retourne toutes les lignes répondant à la condition de filtre.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value filter</td>
<td>Oui</td>
<td>Oui</td>
<td>Oui</td>
<td>Retourne uniquement les colonnes ayant une valeur spécifique.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Les explications concernant l’usage des filtres HBase listés ci-dessus peuvent être modifiées par Apache dans son projet Apache HBase. Afin de comprendre entièrement comment utiliser ces filtres HBase, il est recommandé de lire la documentation Apache HBase.

• Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

• L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à
La connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez [Connexion à une distribution Hadoop personnalisée](#) à la page 1677.

<table>
<thead>
<tr>
<th>HBase version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hadoop version of the distribution</td>
<td>Cette liste s'affiche uniquement lorsque vous avez sélectionné Custom dans la liste des distributions à connecter à un cluster n'étant pas officiellement supporté par le Studio. Dans cette situation, sélectionnez la version de Hadoop de ce cluster personnalisé, Hadoop 1 ou Hadoop 2.</td>
</tr>
<tr>
<td>Zookeeper quorum</td>
<td>Saisissez le nom ou l'URL du service Zookeeper utilisé pour coordonner les transactions entre votre Studio et votre base de données. Notez que, lorsque vous configurez Zookeeper, vous pouvez avoir besoin de configurer explicitement la propriété zookeeper.znode.parent pour définir le chemin vers le nœud znode racine contenant tous les znodes créés et utilisés par votre base de données. Cochez la case la case Set Zookeeper znode parent afin de définir cette propriété.</td>
</tr>
<tr>
<td>Zookeeper client port</td>
<td>Saisissez le numéro du port d'écoute client du service Zookeeper que vous utilisez.</td>
</tr>
</tbody>
</table>
| Use kerberos authentication | Si la base de données choisie utilise la sécurité Kerberos, cochez cette case puis saisissez le nom des principaux dans les champs affichés. Vous pouvez trouver cette information dans le fichier hbase-site.xml du cluster utilisé.  
  - Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l'autheintification par ticket MapR en plus ou comme une alternative en suivant les explications dans [Connexion sécurisée à MapR](#) à la page 1745.  
  Gardez à l'esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.  
  
  Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case **Use a keytab to authenticate**. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d'accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où
s'exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser.

<table>
<thead>
<tr>
<th><strong>Schema et Edit schema</strong></th>
</tr>
</thead>
</table>
| Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th><strong>Built-In</strong></th>
<th>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et il est stocké dans le <strong>Repository</strong>. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Set table Namespace mappings</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisissez la chaîne de caractères à utiliser pour construire le mapping entre une table Apache HBase table et une table MapR. Pour plus d'informations concernant la syntaxe valide à utiliser, consultez <a href="http://doc.mapr.com/display/MapR40x/MappingTable+Namespace+Between+ApacheHBase+Tables+and+MapR+Tables">http://doc.mapr.com/display/MapR40x/MappingTable+Namespace+Between+ApacheHBase+Tables+and+MapR+Tables</a> (en anglais).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Table name</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisissez le nom de la table de la base de données de laquelle vous souhaitez extraire les colonnes.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Define a row selection</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case et, dans les champs <strong>Start row</strong> et <strong>End row</strong>, saisissez les Row Keys correspondants afin de spécifier la plage de lignes que vous souhaitez faire extraire par le composant.</td>
</tr>
</tbody>
</table>
À la différence des filtres que vous pouvez définir à l’aide de l’option **Is by filter** nécessitant le chargement de tous les enregistrements avant de pouvoir filtrer les lignes à utiliser, cette fonctionnalité vous permet de sélectionner directement les lignes qui vous intéressent.

| **Mapping** | Renseignez cette table afin de mapper les colonnes de la table à utiliser avec les colonnes du schéma défini pour le flux de données à traiter. |
| **Die on error** | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**. |

### Advanced settings

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |
| **Properties** | Si vous devez utiliser la configuration personnalisée pour votre base de données, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) vont écraser les propriétés utilisées par le Studio Talend. Par exemple, vous devez définir la valeur de la propriété **dfs.replication** à 1 pour la configuration de base de données. Vous devez ensuite ajouter une ligne à la table à l’aide du bouton **[+]** et saisir le nom et la valeur de cette propriété dans la ligne. **Remarque :** Cette table est indisponible lorsque vous utilisez une connexion existante en cochant la case **Using an existing connection** dans la vue **Basic settings.** |
| **Is by filter** | Cochez cette case pour utiliser des filtres HBase afin d’effectuer une sélection granulaire fine depuis votre base de données, comme une sélection de clés ou de valeurs, selon des expressions régulières. Une fois cette case cochée, la table **Filter** utilisée pour définir les conditions de filtre devient disponible. Cette fonctionnalité tire parti des filtres fournis par HBase et sujets aux contraintes présentées dans la documentation Apache HBase. Un niveau avancé de connaissances de HBase est requis pour que les utilisateurs puissent utiliser pleinement ces filtres. |
| **Logical operation** | Sélectionnez l’opérateur à utiliser pour définir la relation logique entre les filtres. Les opérateurs disponibles sont : |
• **And** : chaque condition de filtre définie doit être satisfaite. Elle représente la relation `FilterList.Operator.MUST_PASS_ALL`.

• **Or** : au moins une des conditions de filtre définie doit être satisfaite. Elle représente la relation : `FilterList.Operator.MUST_PASS_ONE`.

### Filter

Cliquez sur le bouton sous cette table pour ajouter autant de lignes que nécessaire. Chaque ligne représente un filtre. Les paramètres que vous devez configurer pour un filtre sont :

- **Filter type** : la liste déroulante présente les types de filtre prédéfinis par HBase. Sélectionnez le type de filtre à utiliser.

- **Filter column** : saisissez le nom de la colonne (qualifier) sur laquelle vous devez appliquer le filtre actif. Ce paramètre est obligatoire selon le type de filtre et le comparateur que vous utilisez. Par exemple, ce paramètre n'est pas utilisé par le type *Row Filter* mais est requis pour le type *Single Column Value Filter*.

- **Filter family** : saisissez la famille de colonne sur laquelle vous devez appliquer le filtre actif. Ce paramètre est obligatoire selon le type de filtre et le comparateur que vous utilisez. Par exemple, ce paramètre n'est pas utilisé par le type *Row Filter* mais est requis pour le type *Single Column Value Filter*.

- **Filter operation** : sélectionnez dans la liste déroulante l'opération à utiliser pour le filtre actif.

- **Filter Value** : saisissez le valeur sur laquelle utiliser l'opérateur sélectionné dans la liste *Filter operation*.

- **Filter comparator type** : sélectionnez le type de comparateur à combiner au filtre que vous utilisez.

Selon le type de filtre (*Filter type*) que vous utilisez, certains paramètres, voire tous les paramètres sont obligatoires. Pour plus d'informations, consultez Filtres HBase à la page 1493.

### Retrieve timestamps

Cochez cette case afin de charger les horodatages d'une colonne HBase dans le flux de données.

- **Retrieve from an HBase column** : sélectionnez la colonne HBase dont les modifications sont suivies, afin de récupérer ses horodatages.

- **Write to a schema column** : sélectionnez la colonne définie dans le schéma pour stocker les horodatages récupérés.

Le type de la colonne doit être *Long*. 
Variables globales

| Global Variables | NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant d’entrée et requiert un composant de sortie.</th>
</tr>
</thead>
</table>
| Prérequis           | Avant de commencer, vérifiez que tous les prérequis de l’IP de Loopback (rebouclage) attendus par votre base de données sont respectés.
La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.
• Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.
• Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque |
Scénario : Echanger des données clients avec HBase

Ce scénario s'applique uniquement aux solutions Talend avec Big Data.

Dans ce scénario, un Job a six composants est utilisé afin d'échanger des données clients avec une base de données HBase.

Ces six composants sont :

- un **tHBaseConnection** : crée une connexion à votre base de données HBase.
- un **tFixedFlowInput** : crée les données à écrire dans votre base HBase. Dans un cas d'utilisation réel, ce composant peut être remplacé par d'autres composants d’entrée, tels que le **tFileInputDelimited**.
- un **tHBaseOutput** : écrit les données qu'il reçoit du composant précédent dans votre base de données HBase.
- un **tHBaseInput** : extrait les colonnes souhaitées de votre base de données HBase.
- un **tLogRow** : présente les résultats d'exécution.
- un **tHBaseClose** : ferme la connexion.

Pour reproduire ce scénario, procédez comme dans les sections suivantes.
Remarque :
Avant de commencer à reproduire le scénario, vérifiez que votre base de données Hbase et votre service Zookeeper ont été installés et configurés correctement. Ce scénario explique uniquement comment utiliser Talend pour effectuer une transaction avec une base de données HBase.

Déposer et relier les composants

Pourquoi et quand exécuter cette tâche
Pour ce faire, procédez comme suit :

Procédure
1. Déposez un tHBaseConnection, un tFixedFlowInput, un tHBaseOutput, un tHBaseInput, un tLogRow et un tHBaseClose de la Palette dans l'espace de modélisation graphique.
2. Cliquez-droit sur le composant tHBaseConnection afin d'ouvrir son menu contextuel et sélectionnez Trigger > On Subjob Ok pour connecter ce composant au tFixedFlowInput.
3. Répétez l'opération afin de créer le lien OnSubjobOk du tFixedFlowInput au tHBaseInput puis au tHBaseClose.
4. Cliquez-droit sur le tFixedFlowInput et sélectionnez Row > Main afin de connecter ce composant au tHBaseOutput.
5. Répétez l'opération pour créer le lien Main du tHBaseInput au tLogRow.

Résultats
Les composants de ce scénario sont placés et connectés entre eux. Vous devez les configurer.

Configurer la connexion

Pourquoi et quand exécuter cette tâche
Pour configurer la connexion à votre service Zookeeper puis à la base HBase souhaitée, procédez comme suit :

Procédure
1. Dans l'espace de modélisation de votre Studio Talend, double-cliquez sur le composant tHBaseConnection pour ouvrir sa vue Component.

![Capture d'écran de l'interface de configuration de tHBaseConnection](image.png)
2. Sélectionnez Hortonworks Data Platform dans la liste HBase version.

3. Dans le champ Zookeeper quorum, saisissez le nom ou l’URL du service Zookeeper que vous utilisez. Dans cet exemple, le nom du service est hbase.


5. Si l’emplacement de Zookeeper znode parent a été défini dans le cluster Hadoop auquel vous vous connectez, cochez la case Set zookeeper znode parent et saisissez la valeur de la propriété dans le champ affiché.

**Configurer l’écriture de données dans la base HBase**

**Pourquoi et quand exécuter cette tâche**

Pour ce faire, procédez comme suit :

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le composant tFixedFlowInput pour ouvrir sa vue Component.

2. Dans cette vue, cliquez sur le bouton [...] à côté du champ Edit schema afin d’ouvrir l’éditeur de schéma.
3. Cliquez trois fois sur le bouton [+] pour ajouter trois lignes, puis, dans la colonne **Column**, renommez respectivement les trois lignes comme suit : *id, name et age*.

4. Dans la colonne **Type**, cliquez sur chacune de ces lignes, puis dans la liste déroulante, sélectionnez le type de données de chaque ligne. Dans ce scénario, sélectionnez *Integer* pour *id* et *age*, *String* pour *name*.

5. Cliquez sur **OK** afin de valider ces modifications et acceptez la propagation proposée par la fenêtre pop-up.

6. Dans la zone **Mode**, sélectionnez **Use Inline Content (delimited file)** pour afficher les champs à éditer.

7. Dans le champ **Content**, saisissez les données délimitées à écrire dans la base de données HBase, séparées par un ;. Dans cet exemple, les données sont :

   1;Albert;23
   2;Alexandre;24
   3;Alfred-Hubert;22
   4;Andre;40
   5;Didier;28
   6;Anthony;35
   7;Artus;32
   8;Catherine;34
   9;Charles;21
   10;Christophe;36
   11;Christian;67
   12;Danniel;54
   13;Elisabeth;58
   14;Emile;32
   15;Gregory;30

8. Double-cliquez sur le composant **tHBaseOutput** afin d’afficher sa vue **Component**.

   **Remarque** :

   Si ce composant n’a pas le même schéma que le composant précédent, une icône d’avertissement apparaît. Dans ce cas, cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent. L’icône d’avertissement disparaît.
9. Cochez la case **Use an existing connection** puis sélectionnez la connexion précédemment configurée. Dans cet exemple, sélectionnez **tHBaseConnection_1**.

10. Dans le champ **Table name**, saisissez le nom de la table à créer dans la base de données HBase. Dans cet exemple, saisissez **customer**.

11. Dans le champ **Action on table**, sélectionnez l’action que vous souhaitez effectuer, dans la liste déroulante. Dans ce scénario, sélectionnez **Drop table if exists and create**. Ainsi, si une table nommée **customer** existe déjà dans la base de données HBase, elle sera effacée avant la création de la nouvelle table.

12. Cliquez sur l’onglet **Advanced settings** afin d’ouvrir la vue correspondante.

13. Dans la table **Family parameters**, ajoutez deux lignes en cliquant deux fois sur le bouton **[+]** et renommez-les respectivement **family1** et **family2**. Laissez les autres colonnes vides. Ces deux colonnes seront créées dans la base de données HBase avec les options de performances par défaut.

   **Remarque :**
   
   La table **Family parameters** est disponible uniquement lorsque l’action sélectionnée dans la liste **Action on table** est de créer une table dans la base de données HBase. Pour plus d’informations concernant la table **Family parameters**, consultez **tHBaseOutput** à la page 1508.

14. Dans la table **Families** de la vue **Basic settings**, saisissez le nom des familles dans la colonne **Family name**, chaque nom correspondant à la colonne contenue par la famille. Dans cet exemple, les colonnes **id** et **age** appartiennent à la famille **family1** et la colonne **name** à la famille **family2**.

   **Remarque :**
   
   Ces familles de colonnes doivent déjà exister dans la base de données HBase à laquelle vous vous connectez ou elles doivent être définies dans la table **Family parameters** de la vue **Advanced settings**.
Configurer l'extraction des données de la base HBase

Pourquoi et quand exécuter cette tâche

Pour ce faire, effectuez les opérations suivantes :

Procédure

1. Double-cliquez sur le tHBaseInput afin d'ouvrir sa vue Component.

2. Cochez la case **Use an existing connection** puis sélectionnez la connexion configurée précédemment. Dans cet exemple, la connexion est **tHBaseConnection_1**.

3. Cliquez sur le bouton [ ] à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

4. Cliquez trois fois sur le bouton [+] pour ajouter trois colonnes et renommez-les respectivement id, name et age dans la colonne **Column**. Cela signifie que vous allez extraire ces trois colonnes de la base HBase.

5. Sélectionnez le type de chaque colonne. Dans cet exemple, sélectionnez **Integer** pour id et age, **String** pour name.

6. Cliquez sur **OK** pour valider ces modifications et acceptez la propagation proposée par la fenêtre pop-up.

7. Dans le champ **Table name**, saisissez le nom de la table dont vous souhaitez extraire les colonnes. Dans ce scénario, la table se nomme *customer*. 

![Diagramme de configuration](image)
8. Dans la table **Mapping**, la colonne **Column** a été automatiquement renseignée, puisque le schéma est défini. Saisissez simplement le nom de chaque famille dans la colonne **Column family**, chaque nom correspondant à la colonne que la famille contient.

9. Double-cliquez sur le composant **tHBaseClose** pour ouvrir sa vue **Component**.

10. Dans le champ **Component List**, sélectionnez la connexion que vous souhaitez fermer. Dans cet exemple, sélectionnez **tHBaseConnection_1**.

**Exécuter le Job**

Pour exécuter ce Job, appuyez sur **F6**.

La vue **Run** s'ouvre automatiquement et vous pouvez voir les résultats d'exécution.

```
For input string: '30 '
5 | Didier | 20
14 | Emile  | 32
3 | Alfred-Hubert | 22
9 | Charles | 21
1 | Albert  | 23
11 | Christian | 67
7 | Artus   | 32
10 | Christophe | 36
8 | Catherine| 34
13 | Elisabeth| 58
14 | Damien  | 54
6 | Anthony | 35
2 | Alexandre| 24
4 | André   | 40
```

Les colonnes souhaitées sont extraites. Vous pouvez donc les traiter selon vos besoins.

En vous connectant à votre base de données HBase, vous pouvez constater que la table **customer** a bien été créée.
tHBaseOutput

Ce composant écrit des colonnes de données dans une base de données HBase.

Le tHBaseOutput reçoit des données du composant précédent, crée une table dans une base de données HBase et écrit les données reçues dans la table HBase créée.

Propriétés du tHBaseOutput Standard

Ces propriétés sont utilisées pour configurer le tHBaseOutput s’exécutant dans le framework de Jobs Standard.

Le composant tHBaseOutput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de connexion à la base de données et stocker les paramètres de connexion configurés dans la vue Basic settings. Pour plus d’informations concernant la configuration et le stockage des paramètres de connexion à la base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>
| Distribution | Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
  • Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la |
configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option **Custom** vous permet de vous connecter à un cluster différente des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend, Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

| HBase version | Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez. |
| **Hadoop version of the distribution** | Cette liste s’affiche uniquement lorsque vous avez sélectionné Custom dans la liste des distributions à connecter à un cluster n’étant pas officiellement supporté par le Studio. Dans cette situation, sélectionnez la version de Hadoop de ce cluster personnalisé, **Hadoop 1** ou **Hadoop 2**. |
| **Zookeeper quorum** | Saisissez le nom ou l’URL du service Zookeeper utilisé pour coordonner les transactions entre votre Studio et votre base de données. Notez que, lorsque vous configurez Zookeeper, vous pouvez avoir besoin de configurer explicitement la propriété zookeeper.znode.parent pour définir le chemin vers le nœud znode racine contenant tous les znodes créés et utilisés par votre base de données. Cochez la case la case Set Zookeeper znode parent afin de définir cette propriété. |
| **Zookeeper client port** | Saisissez le numéro du port d’écoute client du service Zookeeper que vous utilisez. |
| **Use kerberos authentication** | Si la base de données choisie utilise la sécurité Kerberos, cochez cette case puis saisissez le nom des principaux dans les champs affichés. Vous pouvez trouver cette information dans le fichier hbase-site.xml du cluster utilisé.  
- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.  
Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.  
Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case Use a keytab to authenticate. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.  
Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser. |
**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

**Set table Namespace mappings**

Saisissez la chaîne de caractères à utiliser pour construire le mapping entre une table Apache HBase table et une table MapR.

Pour plus d’informations concernant la syntaxe valide à utiliser, consultez http://doc.mapr.com/display/MapR40x/Mapping+Table+Namespace+Between+Apache+HBase+Tables+and+MapR+Tables (en anglais).

**Table name**

Saisissez le nom de la table HBase que vous souhaitez créer.

**Action on table**

Sélectionnez l’action requise pour créer une table HBase.

**Custom Row Key**

Cochez cette case pour utiliser les clés de lignes personnalisées. Une fois coché, le champ correspondant s’affiche. Saisissez ensuite la clé de ligne.
personnalisée pour indexer les lignes de la table HBase en cours de création.

Par exemple, vous pouvez saisir "France"+Numeric.sequence("sl",1,1) pour produire la série de clés de lignes personnalisées : France1, France2, France3, etc.

| Families | Renseignez cette table afin de mapper les colonnes de la table à utiliser avec les colonnes du schéma défini pour le flux de données à traiter.

La colonne Column de cette table est automatiquement renseignée une fois le schéma défini. Dans la colonne Family name, saisissez les familles de colonnes que vous souhaitez créer ou utiliser pour grouper les colonnes dans la colonne Column. Pour plus d’informations concernant une famille de colonnes, consultez la documentation Apache à l’adresse suivante : Column families (en anglais).

| Custom timestamp column | Sélectionnez une colonne Long de votre schéma afin de fournir les horodatages pour les colonnes HBase à créer ou à mettre à jour par le tHBaseOutput.

| Die on error | Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

**Advanced settings**

| Use batch mode | Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.

| Batch size | Spécifiez le nombre d’enregistrements à traiter dans chaque lot.

Ce champ est disponible uniquement lorsque la case Use batch mode est cochée.

| Properties | Si vous devez utiliser la configuration personnalisée pour votre base de données, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) vont écraser les propriétés utilisées par le Studio Talend.

Par exemple, vous devez définir la valeur de la propriété dfs.replication à 1 pour la configuration de base de données. Vous devez ensuite ajouter une ligne à la table à l’aide du bouton [+] et saisir le nom et la valeur de cette propriété dans la ligne.

- **Remarque :**
  
  Cette table est indisponible lorsque vous utilisez une connexion existante en cochant la case Using an existing connection dans la vue Basic settings.

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
Family parameters

Saisissez les noms et, si nécessaire, les options personnalisées de performance des familles de colonnes à créer. Ces options sont des attributs définis par le modèle de données HBase. Pour plus d’informations concernant ces options, consultez la documentation de Apache HBase.

Remarque :
Le paramètre Compression type vous permet de sélectionner le format de compression des données de sortie.

Variables globales

Global Variables

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de fin et nécessite un composant d’entrée.</th>
</tr>
</thead>
</table>

| Prérequis | Avant de commencer, vérifiez que tous les prérequis de l’IP de Loopback (rebouclage) attendus par votre base de données sont respectés.

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client. |
MapR. Pour plus d'informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n'ajoutez pas de librairie, il est possible que vous rencontriez l'erreur suivante : no MapRClient in java.library.path.

- Configurez l'argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]** dans le menu **Window**. Cet argument fournit au studio le chemin d'accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d'utiliser entièrement l'aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d'informations concernant l'installation d'une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

**Scénario associé**

Pour un scénario relatif à la version Standard du tHBaseOutput, consultez Scénario : Echanger des données clients avec HBase à la page 1501.
tHCatalogInput

Ce composant lit des données d'une base de données Hive spécifiée gérée par HCatalog et d'envoyer les données au composant suivant.

Propriétés du tHCatalogInput Standard

Ces propriétés sont utilisées pour configurer le tHCatalogInput s'exécutant dans le framework de Jobs Standard.

Le composant tHCatalogInput Standard appartient à la famille .

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Property type | Peut être **Built-in** ou **Repository**.  
**Built-in** : Propriétés utilisées ponctuellement.  
**Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées. |
|---|---|
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.  
Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
• **View schema** : sélectionnez cette option afin de voir le schéma.  
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| | **Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du *Studio Talend*. |

Distribution

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont
Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’im porter les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>HCatalog version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
</table>
| Templeton hostname | Renseignez ce champ avec l’URL du service Web Templeton.  

**Remarque :**
Templeton est une API de Service Web pour HCatalog. Elle a été renommée WwebHCat par la communauté Apache. Ce service permet d’accéder à HCatalog et aux éléments Hadoop relatifs, comme Pig. Pour plus d’informations concernant Templeton (WebHCat), consultez https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat (en anglais). |
| Templeton port | Renseignez ce champ avec le port de l’URL du service Web Templeton. Par défaut, cette valeur est 50111.  

**Remarque :**
Templeton est une API de Service Web pour HCatalog. Elle a été renommée WwebHCat par la communauté Apache. Ce service permet d’accéder à HCatalog et aux éléments Hadoop relatifs, comme Pig. Pour plus d’informations concernant Templeton (WebHCat), consultez https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat (en anglais). |
| Use kerberos authentication | Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.  

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745. |
Gardez à l'esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

| Use a keytab to authenticate | Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend. Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser. |
| Database | Base de données dans laquelle les tables gérées par HCatalog sont stockées. Par défaut, cette base de données est la base Hive nommée *default*. |
| Table | Renseignez ce champ afin d’effectuer des opérations sur une ou plusieurs table(s) dans la base de données spécifiée. |
| Partition | Renseignez ce champ pour spécifier une ou plusieurs partition(s) pour l’opération de partition sur la table spécifiée. Lorsque vous spécifiez différentes partitions, utilisez une virgule pour séparer chaque groupe de deux partitions. Utilisez également des guillemets doubles afin d’entourer la chaîne de caractères de la partition. Si vous lisez une table non partitionnée, laissez ce champ vide.  

**Remarque :** Pour plus d’informations concernant les partitions, consultez [https://cwiki.apache.org/Hive/](https://cwiki.apache.org/Hive/) (en anglais). |
| Username | Renseignez ce champ avec le nom d’utilisateur de connexion à la base de données Hive. |
| Die on error | Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur. |
**Advanced settings**

<table>
<thead>
<tr>
<th><strong>Row separator</strong></th>
<th>Saisissez le séparateur utilisé pour identifier la fin d'une ligne.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field separator</strong></td>
<td>Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.</td>
</tr>
<tr>
<td><strong>Custom encoding</strong></td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.</td>
</tr>
</tbody>
</table>
| **Hadoop properties** | Le Studio Talend utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.  
  • Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste Property type, vous passiez de Repository à Built-in.  
  Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur http://hadoop.apache.org/docs en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :  
  • Généralement, les propriétés relatives à HDFS peuvent être trouvées dans le fichier hdfs-default.xml correspondant à votre distribution, comme par exemple http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais).  
  • Apache fournit également une page listant toutes les propriétés relatives à Hive : http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais). |
| **Retrieve the HCatalog logs** | Cochez cette case pour récupérer les fichiers de log générés durant les opérations de HCatalog. |
| **Standard Output Folder** | Renseignez ce champ avec le chemin d’accès où sont stockés les fichiers de log.  
  **Remarque :**  
  Ce champ est activé lorsque vous cochez la case Retrieve the HCatalog logs. |
Error Output Folder
Renseignez ce champ avec le chemin d'accès où sont stockés les fichiers de log d'erreurs.

**Remarque :**
Ce champ est activé lorsque vous cochez la case *Retrieve the HCatalog logs.*

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.

### Global Variables

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend.*

### Utilisation

**Règle d’utilisation**
Ce composant est généralement utilisé en tant que composant de début dans un Job.

**Prérequis**
La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*.
La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le *Studio Talend* et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées `MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native`. Par exemple, pour Windows, la bibliothèque est `lib\MapRClient.dll` dans le fichier JAR du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : `no MapRClient in java.library.path.`
• Configurez l’argument `-Djava.library.path`, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

| Limitation | Lorsque la case Use kerberos authentication est cochée, le composant ne fonctionne pas avec la JVM IBM. |

### Scénario associé

Pour un scénario associé, consultez Scénario : Gestion de table HCatalog dans Hortonworks Data Platform à la page 1537.
Ce composant lit des données directement de HDFS et écrit ces données dans une table établie gérée par HCatalog.

**Propriétés du tHCatalogLoad Standard**

Ces propriétés sont utilisées pour configurer le tHCatalogLoad s'exécutant dans le framework de Jobs Standard.

Le composant tHCatalogLoad Standard appartient à la famille .

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Basic settings**

| Property type | Peut être **Built-in** ou **Repository**.  
**Built-in** : Propriétés utilisées ponctuellement.  
**Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées. |  
---|---|  
**Distribution** | Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
• Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).  
• L’option **Custom** vous permet de vous connecter à un cluster différente des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**.  
1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,  
2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution |
Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th><strong>HCatalog version</strong></th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Templeton hostname</strong></td>
<td>Renseignez ce champ avec l’URL du service Web Templeton.</td>
</tr>
</tbody>
</table>

**Remarque :**
Templeton est une API de Service Web pour HCatalog. Elle a été renommée WwebHCat par la communauté Apache. Ce service permet d’accéder à HCatalog et aux éléments Hadoop relatifs, comme Pig. Pour plus d’informations concernant Templeton (WebHCat), consultez https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat (en anglais).
| Templeton port | Renseignez ce champ avec le port de l'URL du service Web Templeton. Par défaut, cette valeur est 50111.  
**Remarque :** Templeton est une API de Service Web pour HCatalog. Elle a été renommée WwebHCat par la communauté Apache. Ce service permet d'accéder à HCatalog et aux éléments Hadoop relatifs, comme Pig. Pour plus d'informations concernant Templeton (WebHCat), consultez https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat (en anglais). |
| Use kerberos authentication | Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le 'principal name' de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d'utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.  
- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l'authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745. Gardez à l'esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d'utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.  
Cette case est disponible ou indisponible selon la distribution d'Hadoop à laquelle vous vous connectez. |
<p>| Use a keytab to authenticate | Cochez la case Use a keytab to authenticate pour vous connecter à un système utilisant Kerberos à l'aide d'un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d'accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend. Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser. |
| Database | Saisissez le nom de la base de données dans laquelle écrire les données. Cette base de données doit exister. |</p>
<table>
<thead>
<tr>
<th>Table</th>
<th>Saisissez le nom de la table dans laquelle écrire les données. Cette table doit exister.</th>
</tr>
</thead>
</table>
| Partition | Renseignez ce champ pour spécifier une ou plusieurs partition(s) pour l'opération de partition sur la table spécifiée. Lorsque vous spécifiez différentes partitions, utilisez une virgule pour séparer chaque groupe de deux partitions. Utilisez également des guillemets doubles afin d’entourer la chaîne de caractères de la partition. Si vous lisez une table non partitionnée, laissez ce champ vide.  

**Remarque :**  
Pour plus d’informations concernant les partitions, consultez https://cwiki.apache.org/Hive/ (en anglais). |
| Username | Renseignez ce champ avec le nom d’utilisateur de connexion à la base de données. |
| File location | Saisissez le chemin absolu pointant vers l’emplacement HDFS d’où sont lues les données. |

**Advanced settings**

| Retrieve the HCatalog logs | Cochez cette case pour récupérer les fichiers de log générés durant les opérations HCatalog. |
| Standard Output Folder | Renseignez ce champ avec le chemin d’accès aux fichiers de log stockés.  

**Remarque :**  
Ce champ est activé lorsque vous sélectionnez Retrieve the HCatalog logs. |
| Error Output Folder | Renseignez ce champ avec le chemin d’accès aux fichiers d’erreurs stockés.  

**Remarque :**  
Ce champ est activé lorsque vous sélectionnez Retrieve the HCatalog logs. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job aussi bien qu’au niveau des composants. |

**Global Variables**

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. |
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

---

### Utilisation

#### Règle d'utilisation

Ce composant est généralement utilisé seul dans un sous-job à un composant.

HCatalog est construit sur le métastore Hive afin de fournir une interface de lecture/écriture pour Pig et MapReduce, afin que ces systèmes puissent utiliser les métadonnées de Hive pour lire et écrire facilement des données dans HDFS.

Pour plus d'informations, consultez la documentation Apache concernant HCatalog : [https://cwiki.apache.org/confluence/display/Hive/HCatalog](https://cwiki.apache.org/confluence/display/Hive/HCatalog) (en anglais).

#### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées `MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native`. Par exemple, pour Windows, la bibliothèque est `lib\MapRClient.dll` dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

  Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : *no MapRClient in java.library.path.*

- Configurez l’argument `−Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.
Scénario associé

Pour un scénario associé, consultez Scénario : Gestion de table HCatalog dans Hortonworks Data Platform à la page 1537.
**tHCatalogOperation**

Ce composant prépare la base de données/table/partition gérée par HCatalog à traiter.

Le tHCatalogOperation gère les données stockées dans la base de données/table/partition Hive gérée par HCatalog.

**Propriétés du tHCatalogOperation Standard**

Ces propriétés sont utilisées pour configurer le tHCatalogOperation s’exécutant dans le framework de Jobs Standard.

Le composant tHCatalogOperation Standard appartient à la famille .

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Property type | Peut être **Built-in** ou **Repository**.  
**Built-in** : Propriétés utilisées ponctuellement.  
**Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées. |

| Distribution | Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).  
- L’option **Custom** vous permet de vous connecter à un cluster différente des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.  
1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou, |
2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l'évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d'utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l'opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d'Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

HCatalog version
 Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.

Templeton hostname
 Renseignez ce champ avec l’URL du service Web Templeton.

Remarque :
Templeton est une API de Service Web pour HCatalog. Elle a été renommée WwebHCat par la communauté Apache. Ce service permet d'accéder à HCatalog et aux éléments Hadoop relatifs, comme Pig. Pour plus d'informations concernant Templeton (WebHCat), consultez...
### Templeton port

Renseignez ce champ avec le port de l'URL du service Web Templeton. Par défaut, cette valeur est 50111. Templeton est une API de services Web pour Hadoop.

**Remarque :**

Templeton est une API de Service Web pour HCatalog. Elle a été renommée WwebHCat par la communauté Apache. Ce service permet d'accéder à HCatalog et aux éléments Hadoop relatifs, comme Pig. Pour plus d’informations concernant Templeton (WebHCat), consultez https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat (en anglais).

### Use kerberos authentication

Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décrochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

### Use a keytab to authenticate

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est **user1** et le principal à utiliser est **guest**. Dans cette situation, assurez-vous que **user1** a les droits de lecture pour le fichier keytab à utiliser.
| Operation on | Sélectionnez un objet dans la liste pour les opérations sur la base de données.  
**Database** : Base de données gérée par HCatalog dans HDFS.  
**Table** : Table gérée par HCatalog dans HDFS.  
**Partition** : Partition définie par l’utilisateur. |
|--------------|----------------------------------------------------------------------------------------------------------|
| Create the table only it doesn’t exist already | Cochez cette case afin d’éviter de créer un doublon lorsque vous créez une table.  
**Remarque** :  
Cette case est activée lorsque vous sélectionnez **Table** dans la liste **Operation on**. |
| Database     | Base de données dans laquelle les tables gérées par HCatalog sont stockées.  
**Remarque** :  
Ce champ est activé lorsque vous sélectionnez **Table** dans la liste **Operation on**. Pour plus d’informations concernant les opérations de Partition, consultez [https://cwiki.apache.org/Hive/](https://cwiki.apache.org/Hive/). |
| Table        | Renseignez ce champ pour effectuer des actions sur une ou plusieurs table(s) dans une base de données ou dans un emplacement HDFS spécifié.  
**Remarque** :  
Ce champ est activé lorsque vous sélectionnez **Table** dans la liste **Operation on**. Pour plus d’informations concernant les opérations de Partition, consultez [https://cwiki.apache.org/Hive/](https://cwiki.apache.org/Hive/). |
| Partition    | Renseignez ce champ pour spécifier une ou plusieurs partition(s) pour l’opération de partition sur une table. Lorsque vous spécifiez plusieurs partitions, utilisez une virgule pour séparer les partitions en groupes de deux. Délimitez la chaîne de caractères relative à la partition en l’entourant de guillemets doubles.  
Si vous lisez une table non partitionnée, laissez ce champ vide.  
**Remarque** :  
Ce champ est activé lorsque vous sélectionnez **Table** dans la liste **Operation on**. Pour plus d’informations concernant les opérations de Partition, consultez [https://cwiki.apache.org/Hive/](https://cwiki.apache.org/Hive/). |
| Username     | Renseignez ce champ en saisissant l’identifiant de l’utilisateur de la base de données.  
**Remarque** :  
Ce champ est activé lorsque vous sélectionnez **Table** dans la liste **Operation on**. Pour plus d’informations concernant les opérations de Partition, consultez [https://cwiki.apache.org/Hive/](https://cwiki.apache.org/Hive/). |
| Database location | Emplacement de la base de données dans HDFS.  
**Remarque** : |
<table>
<thead>
<tr>
<th>Database description</th>
<th>Description de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Ce champ est activé lorsque vous sélectionnez <strong>Database</strong> dans la liste <strong>Operation on</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Create an external table</th>
<th>Cochez cette case pour créer une table externe avec un chemin d'accès alternatif défini dans le champ <strong>Set HDFS location</strong>, dans la vue <strong>Advanced settings</strong>. Pour plus d’informations concernant la création d’une table externe, consultez <a href="https://cwiki.apache.org/Hive/">https://cwiki.apache.org/Hive/</a> (en anglais).</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Cette case est activée lorsque vous sélectionnez <strong>Table</strong> dans la liste <strong>Operation on</strong> et <strong>Create</strong> ou <strong>Drop and create</strong> ou <strong>Drop if exist and create</strong> dans la liste <strong>Operation</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Format</th>
<th>Sélectionnez un format de fichier dans la liste, afin de spécifier le format de la table externe que vous souhaitez créer.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>TEXTFILE</strong> : Fichiers texte.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>L’option <strong>RCFILE</strong> est disponible uniquement depuis Hive 0.6.0. Cette liste est activée uniquement lorsque vous sélectionnez <strong>Table</strong> dans la liste <strong>Operation on</strong> et <strong>Create/Drop and create/Drop if exist and create</strong> dans la liste <strong>Operation</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set partitions</th>
<th>Cochez cette case pour définir le schéma de la partition en cliquant sur <strong>Edit schema</strong>, à droite de la case <strong>Set partitions</strong>. Le schéma de la partition peut être soit local, soit distant dans le <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Cette case est activée lorsque vous sélectionnez <strong>Table</strong> dans la liste <strong>Operation on</strong> et <strong>Create, Drop and create</strong> ou <strong>Drop if exist and create</strong> dans la liste <strong>Operation</strong>. Vous devez suivre les règles d’utilisation de schémas de partition dans les tables gérées par HCatalog. Pour plus d’informations concernant les règles d’utilisation des schémas de partitions, consultez <a href="http://incubator.apache.org/hcatalog/">http://incubator.apache.org/hcatalog/</a> (en anglais).</td>
</tr>
</tbody>
</table>
### Built-in
Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository
Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

#### Set the user group to use
Cochez cette case pour spécifier le groupe de l'utilisateur.

**Remarque :**
Cette case est activée lorsque vous sélectionnez Database dans la liste Operation on et Drop, Drop if exist, Drop and create ou Drop if exist and create dans la liste Operation. Par défaut, la valeur de ce champ est root. Pour plus d’informations concernant le groupe d’utilisateurs dans le serveur, contactez votre administrateur système.

#### Option
Sélectionnez une clause pour lorsque vous supprimez une base de données.

**Remarque :**
Cette liste est activée lorsque vous sélectionnez Drop, Drop if exist, Drop and create ou Drop if exist and create dans la liste Operation. Pour plus d’informations concernant l’opération Drop sur les bases de données, consultez https://cwiki.apache.org/Hive/ (en anglais).

#### Set the permissions to use
Cochez cette case pour spécifier les permissions nécessaires pour l’opération sélectionnée dans la liste Operation.

**Remarque :**
Cette case est activée lorsque vous sélectionnez Drop, Drop if exist, Drop and create ou Drop if exist and create dans la liste Operation. Par défaut, la valeur de ce champ est rwxrw-r-x. Pour plus d’informations concernant les permissions, contactez votre administrateur système.

#### Set File location
Saisissez le répertoire dans lequel seront stockées les données partitionnées.

**Remarque :**
Cette case est activée lorsque vous sélectionnez Partition dans la liste Operation on et Create, Drop and create ou Drop if exist and create dans la liste Operation. Pour plus d’informations concernant le stockage des données partitionnées dans HDFS, consultez https://cwiki.apache.org/confluence/display/Hive/HCatalog (en anglais).
## Die on error

Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

### Advanced settings

#### Comment

Renseignez ce champ à l’aide de commentaires concernant la table à créer.

**Remarque :**

Ce champ est activé lorsqu vous sélectionnez **Table** dans la liste **Operation on** et **Create, Drop and create ou Drop if exist and create** dans la liste **Operation**, dans la vue **Basic settings**.

#### Set HDFS location

Cochez cette case pour spécifier un emplacement HDFS où sauvegarder la table à créer. Décochez cette case afin de sauvegarder la table dans le répertoire du warehouse défini dans la clé **hive.metastore.warehouse.dir** dans le fichier de configuration Hive, **hive-site.xml**.

**Remarque :**

Cette case est activée lorsque vous sélectionnez **Table** dans la liste **Operation on** et **Create, Drop and create ou Drop if exist and create** dans la liste **Operation**, dans la vue **Basic settings**. Pour plus d’informations concernant la sauvegarde de données dans HDFS, consultez [https://cwiki.apache.org/Hive/](https://cwiki.apache.org/Hive/) (en anglais).

#### Set row format(terminated by)

Cochez cette case pour utiliser et définir les formats de lignes lorsque vous créez une table.

- **Field** : Cochez cette case pour utiliser le format **Field**. La valeur par défaut de ce champ est "\u0001". Vous pouvez également spécifier un caractère personnalisé.

- **Collection Item** : Cochez cette case pour utiliser le format **Collection Item**. La valeur par défaut de ce champ est "\u0002". Vous pouvez également spécifier un caractère personnalisé.

- **Map Key** : Cochez cette case afin d’utiliser le format **Map Key**. La valeur par défaut de ce champ est "\u0003". Vous pouvez également spécifier un caractère personnalisé.

- **Line** : Cochez cette case pour utiliser le format **Line**. La valeur par défaut de ce champ est "\n". Vous pouvez également spécifier un caractère personnalisé

**Remarque :**

Cette case est activée lorsque vous sélectionnez **Table** dans la liste **Operation on** et **Create, Drop and create ou Drop if exist and create** dans la liste **Operation**, dans la vue **Basic settings**. Pour plus d’informations concernant les formats de ligne dans les tables...
### Properties

Cliquez sur le bouton [+] pour ajouter une ou plusieurs ligne(s) afin de définir les propriétés de la table. Les propriétés de la table vous permettent de personnaliser la définition de la table à l'aide de vos propres paires de clés de métadonnées/valeurs. Assurez-vous que les valeurs dans les colonnes *Key* et *Value* sont bien entourées de guillemets doubles.

**Remarque :**
Ce tableau est activé lorsque vous sélectionnez *Database* ou *Table* dans la liste *Operation on* et *Create, Drop and create* ou *Drop if exist and create* dans la liste *Operation*, dans la vue *Basic settings*. Pour plus d’informations concernant les propriétés des tables, consultez [https://cwiki.apache.org/Hive](https://cwiki.apache.org/Hive) (en anglais).

### Retrieve the HCatalog logs

Cochez cette case pour récupérer les fichiers de log générés durant les opérations de HCatalog.

### Standard Output Folder

Renseignez ce champ avec le chemin d’accès où sont stockés les fichiers de log.

**Remarque :**
Ce champ est activé lorsque vous cochez la case *Retrieve the HCatalog logs*.

### Error Output Folder

Renseignez ce champ avec le chemin d’accès où sont stockés les fichiers de log d’erreurs.

**Remarque :**
Ce champ est activé lorsque vous cochez la case *Retrieve the HCatalog logs*.

### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

### Global Variables

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est généralement utilisé seul dans un sous-job à un composant. HCatalog est construit sur le métastore Hive afin de fournir une interface de lecture/écriture pour Pig et MapReduce, afin que ces systèmes puissent utiliser les métadonnées de Hive pour lire et écrire facilement des données dans HDFS. Pour plus d’informations, consultez la documentation Apache concernant HCatalog : <a href="https://cwiki.apache.org/confluence/display/Hive/HCatalog">https://cwiki.apache.org/confluence/display/Hive/HCatalog</a> (en anglais).</th>
</tr>
</thead>
</table>

| Prérequis | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.  
- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.  
- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR. Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez. |
|---------------------|--------------------------------------------------------------------------------------------------------|

| Limitation | Lorsque la case Use kerberos authentication est cochée, le composant ne fonctionne pas avec la JVM IBM. |
**Scénario : Gestion de table HCatalog dans Hortonworks Data Platform**

Ce scénario s'applique uniquement aux solutions Talend avec Big Data.

Ce scénario décrit un Job à six composants effectuant les opérations les plus courantes pour la gestion de tables HCatalog dans Hortonworks Data Platform. Les sous-sections de ce scénario traitent les opérations suivantes sur les bases de données :

- Créer une table dans la base de données dans HDFS ;
- Écrire des données dans une table gérée par HCatalog ;
- Écrire des données dans la table partitionnée, à l'aide de `tHCatalogLoad` ;
- Lire des données d'une table gérée par HCatalog ;
- Écrire en sortie les données lues de la table dans HDFS.

**Remarque** :


**Construire le Job**

**Procédure**

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique :
   - un `tHCatalogOperation`, un `tHCatalogLoad`, un `tHCatalogInput`, un `tHCatalogOutput`, un `tFixedFlowInput` et un `tFileOutputDelimited`. 

2. Cliquez-droit sur le **tHCatalogOperation** afin de le relier au **tFixedFlowInput** à l'aide d’un lien **Trigger>OnSubjobOk**.
3. Cliquez-droit sur le **tFixedFlowInput** afin de le connecter au **tHCatalogOutput** à l’aide d’un lien **Row>** **Main**.
4. Cliquez-droit sur le composant **tFixedFlowInput** pour le relier au **tHCatalogLoad** à l’aide d’un lien **Trigger>OnSubjobOk**.
5. Cliquez-droit sur le **tHCatalogLoad** pour le connecter au **tHCatalogInput** à l’aide d’un lien **Trigger>OnSubjobOk**.
6. Cliquez-droit sur le composant **tHCatalogInput** afin de le relier au **tFileOutputDelimited** à l’aide d’un lien **Row>Main**.

**Créer une table dans HDFS**

**Procédure**

1. Double-cliquez sur le **tHCatalogOperation** pour ouvrir sa vue **Basic settings**.
2. Cliquez sur le bouton [...] à côté de **Edit schema** pour définir le schéma de la table à créer.

![screenshot of the thCatalogOperation interface](image)

2. Cliquez sur le bouton [...] à côté de **Edit schema** pour définir le schéma de la table à créer.

3. Cliquez sur le bouton [+] pour ajouter au moins une colonne au schéma puis cliquez sur **OK** lorsque vous avez terminé la configuration. Dans ce scénario, les colonnes ajoutées au schéma sont : *name*, *country* et *age*.

4. Renseignez le champ **Templeton hostname** avec l’URL du service Web Templeton que vous utilisez. Dans ce scénario, saisissez “192.168.0.131”.

5. Renseignez le numéro du port dans le champ **Templeton port** pour le nom du port renseigné dans le champ **Templeton hostname**. Par défaut, la valeur de ce champ est “50111”.

6. Sélectionnez **Table** dans la liste **Operation on** et **Drop if exist and create** dans la liste **Operation** afin de créer une table dans HDFS.

7. Dans le champ **Database**, saisissez le nom d’une base de données existante dans HDFS. Dans ce scénario, le nom de la base de données est “talend”.
8. Dans le champ **Table**, saisissez le nom de la table à créer. Dans ce scénario, le nom de la table est "Customer".


10. Cochez la case **Set the user group to use** afin de spécifier le groupe de l’utilisateur. Le groupe par défaut est "root". Vous devez paramétrer la valeur de ce champ selon vos pratiques réelles.

11. Cochez la case **Set the permissions to use** afin de spécifier les permissions de l’utilisateur. La valeur par défaut de ce champ est "rwxrwxr-x".

12. Cochez la case **Set partitions** afin d’activer le schéma de partition

13. Cliquez sur le bouton **Edit schema [...]** à côté de la case **Set partitions**, afin de définir le schéma de la partition.

14. Cliquez sur le bouton [+] pour ajouter une colonne au schéma et cliquez sur **OK** lorsque vous avez terminé la configuration. Dans ce scénario, la colonne ajoutée au schéma est nommée : *match_age*.

### Écrire des données dans la table existante

**Procédure**

1. Double-cliquez sur le composant **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour définir un schéma semblable à celui défini dans le **tHCatalogOperation**.

3. Dans le champ **Number of rows**, saisissez l’entier 8.

4. Sélectionnez l’option **Use Inline Table** dans la zone **Mode**.

5. Cliquez sur le bouton [+] pour ajouter de nouvelles lignes à la table.

6. Double-cliquez sur le **tHCatalogOutput** pour ouvrir sa vue **Basic settings**.
7. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma défini dans le composant précédent.


10. Sélectionnez **Overwrite** dans la liste **Action**.

11. Dans le champ **Templeton hostname**, saisissez l’URL du service Web Templeton que vous utilisez. Dans ce scénario, saisissez "192.168.0.131".

12. Dans le champ **Templeton port**, saisissez le numéro du port défini dans le champ **Templeton hostname**. Par défaut, la valeur de ce champ est "50111"

13. Dans les champs **Database**, **Table** et **Username**, saisissez les mêmes valeurs que dans le composant `tHCatalogOperation`.

14. Dans le champ **Partition**, saisissez "`match_age=27`".


**Écrire des données dans la table partitionnée, via le tHCatalogLoad**

**Procédure**

1. Double-cliquez sur le `tHCatalogLoad` afin d’ouvrir sa vue **Basic settings**.
2. Dans le champ **Partition**, saisissez "**match_age=26**".
3. Configurez les autres paramètres comme dans le **thCatalogOperation**.

**Lire des données de la table dans HDFS**

**Procédure**

1. Double-cliquez sur le **thCatalogInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton [...] à côté du champ **Edit schema** afin de définir le schéma de la table à lire depuis la base de données.
3. Cliquez sur le bouton [+] pour ajouter au moins une colonne au schéma. Dans ce scénario, les colonnes ajoutées au schéma sont nommées **age** et **name**, respectivement.

4. Dans le champ **Partition**, saisissez “**match_age=26**”.

5. Configurez les autres paramètres comme dans le **tHCatalogOperation**.

**Écrire dans la console les données lues de la table dans HDFS**

**Procédure**

1. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur **Sync columns** afin de récupérer le schéma défini dans le composant précédent.

3. Sélectionnez l’option **Table (print values in cells of a table)** dans la zone **Mode**.

**Exécuter le Job**

Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job et appuyez sur **F6** pour l’exécuter.
Les données de la table restreinte lues depuis HDFS sont affichées dans la console.

Contents of directory /user/hdp/Customer

Goto: /user/hdp/Customer go

Go to parent directory

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Replication</th>
<th>Block Size</th>
<th>Modification Time</th>
<th>Permission</th>
<th>Owner</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer.csv</td>
<td>file</td>
<td>1.04 KB</td>
<td>3</td>
<td>64 MB</td>
<td>2012-04-18 16:16</td>
<td>rw-r--r--</td>
<td>hdp</td>
<td>hdfs</td>
</tr>
</tbody>
</table>

Go back to DFS home

Local logs

Log directory

This is Apache Hadoop release 1.0.2

Cliquez sur le lien Customer.csv pour visualiser le contenu de la table créée.
File: /user/hdp/Customer/Customer.csv

Go to: /user/hdp/Customer  go

Go back to dir listing
Advanced view/download options

27; Tunisia; Abu
31; Senegal; Aneika
26; Morocco; Karim
33; Senegal; Drogba
27; Japan; Ito
26; Algeria; Zeid
33; Japan; Hidetoshi
26; Lebanon; Nasri
27; Tunisia; Abu
31; Senegal; Aneika
26; Morocco; Karim
33; Senegal; Drogba
27; Japan; Ito
26; Algeria; Zeid
33; Japan; Hidetoshi
26; Lebanon; Nasri
27; Tunisia; Abu
31; Senegal; Aneika
26; Morocco; Karim
33; Senegal; Drogba
27; Japan; Ito
26; Algeria; Zeid
33; Japan; Hidetoshi
26; Lebanon; Nasri
27; Tunisia; Abu
31; Senegal; Aneika

Download this file
Tell this file

Chunk size to view (in bytes, up to file's DFS block size): 32768  Refresh
**tHCatalogOutput**

Ce composant reçoit des données du flux d’entrée et les écrit dans une table gérée par HCatalog.

**Propriétés du tHCatalogOutput Standard**

Ces propriétés sont utilisées pour configurer le tHCatalogOutput s’exécutant dans le framework de Jobs Standard.

Le composant tHCatalogOutput Standard appartient à la famille .

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

### Basic settings

| Property type | Peut être *Built-in* ou *Repository*.  
*Built-in* : Propriétés utilisées ponctuellement.  
*Repository* : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Distribution

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option **Custom** vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend. Talend** et
sa Communauté fournissent l'opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d'Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>HCatalog version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
</table>
| Use kerberos authentication | Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.  
  - Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.
  
  Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.
  
  Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez. |
| Use a keytab to authenticate | Cochez la case Use a keytab to authenticate pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab.  
  Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend. |
Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser.

### NameNode URI

Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée *masternode* comme NameNode, son emplacement est `hdfs://masternode:portnumber`. Si vous utilisez WebHDFS, l’emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être `swebhdfs` et vous devez utiliser un `tLibraryLoad` dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

### File name

Parcourez votre système ou saisissez le chemin d’accès au fichier dans lequel vous souhaitez écrire les données. Ce fichier est automatiquement créé s’il n’existe pas.

### Action

Sélectionnez une opération sur la base de données dans HDFS :

- **Create**: Créer un fichier avec des données à l’aide du nom de fichier défini dans le champ *File Name*.
- **Overwrite**: Écraser les données dans le fichier spécifié dans le champ *File Name*.
- **Append**: Insérer les données dans le fichier spécifié dans le champ *File Name*. Le fichier spécifié est automatiquement créé s’il n’existe pas.

### Templeton hostname

Renseignez ce champ avec l’URL du service Web Templeton.

**Remarque :**

Templeton est une API de Service Web pour HCatalog. Elle a été renommée WwebHCat par la communauté Apache. Ce service permet d’accéder à HCatalog et aux éléments Hadoop relatifs, comme Pig. Pour plus d’informations concernant Templeton (WebHCat), consultez [https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat](https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat) (en anglais).

### Templeton port

Renseignez ce champ avec le port de l’URL du service Web Templeton. Par défaut, cette valeur est 50111.

**Remarque :**

Templeton est une API de Service Web pour HCatalog. Elle a été renommée WwebHCat par la communauté Apache. Ce service permet d’accéder à HCatalog et aux éléments Hadoop relatifs, comme Pig. Pour plus d’informations concernant Templeton (WebHCat), consultez [https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat](https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat) (en anglais).
<table>
<thead>
<tr>
<th><strong>tHCatalogOutput</strong></th>
<th>![tHCatalogOutput](<a href="https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat">https://cwiki.apache.org/confluence/display/Hive/WebHCat+UsingWebHCat</a> (en anglais).)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Renseignez ce champ pour spécifier une base de données existant dans HDFS.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Renseignez ce champ pour spécifier une table existant dans HDFS.</td>
</tr>
<tr>
<td><strong>Partition</strong></td>
<td>Renseignez ce champ pour spécifier une ou plusieurs partition(s) pour l’opération de partition sur la table spécifiée. Lorsque vous spécifiez différentes partitions, utilisez une virgule pour séparer chaque groupe de deux partitions. Utilisez également des guillemets doubles afin d’entourer la chaîne de caractères de la partition. Si vous lisez une table non partitionnée, laissez ce champ vide. <strong>Remarque :</strong> Pour plus d’informations concernant les partitions, consultez <a href="https://cwiki.apache.org/Hive/">https://cwiki.apache.org/Hive/</a> (en anglais).</td>
</tr>
<tr>
<td><strong>Username</strong></td>
<td>Renseignez ce champ avec le nom d’utilisateur de connexion à la base de données.</td>
</tr>
<tr>
<td><strong>File location</strong></td>
<td>Renseignez ce champ avec le chemin d’accès à l’emplacement où est stocké le fichier des données source.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| **Row separator** | Saisissez le séparateur utilisé pour identifier la fin d’une ligne. |
| **Field separator** | Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées. |
| **Custom encoding** | Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données. |
| **Hadoop properties** | Le **Studio Talend** utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut. |
• Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste Property type, vous passiez de Repository à Built-in.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur http://hadoop.apache.org/docs en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :

• Généralement, les propriétés relatives à HDFS peuvent être trouvées dans le fichier hdfs-default.xml correspondant à votre distribution, comme par exemple http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais).

• Apache fournit également une page listant toutes les propriétés relatives à Hive : http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais).

---

Retrieve the HCatalog logs

Cochez cette case pour récupérer les fichiers de log générés durant les opérations de HCatalog.

Standard Output Folder

Renseignez ce champ avec le chemin d’accès où sont stockés les fichiers de log.

ℹ️ Remarque :
Ce champ est activé lorsque vous cochez la case Retrieve the HCatalog logs.

Error Output Folder

Renseignez ce champ avec le chemin d’accès où sont stockés les fichiers de log d’erreurs.

ℹ️ Remarque :
Ce champ est activé lorsque vous cochez la case Retrieve the HCatalog logs.

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

---

Variables globales

Variables globales

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

#### Règle d’utilisation

Ce composant de sortie est généralement utilisé avec un composant d’entrée.

HCatalog est construit sur le métastore Hive afin de fournir une interface de lecture/écriture pour Pig et MapReduce, afin que ces systèmes puissent utiliser les métdonnes de Hive pour lire et écrire facilement des données dans HDFS.

Pour plus d’informations, consultez la documentation Apache concernant HCatalog : [https://cwiki.apache.org/confluence/display/Hive/HCatalog](https://cwiki.apache.org/confluence/display/Hive/HCatalog) (en anglais).

#### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend.

La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées `MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native`. Par exemple, pour Windows, la bibliothèque est `lib\MapRClient.dll` dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : `no MapRClient in java.library.path`.

- Configurez l’argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue Run/Debug de la boîte de dialogue **[Preferences]** dans le menu **Window**. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.
Scénario associé

Pour un scénario associé, consultez Scénario : Gestion de table HCatalog dans Hortonworks Data Platform à la page 1537.
tHDFSCCompare

Ce composant compare deux fichiers dans HDFS et, selon le schéma en lecture seule, génère une ligne de flux présentant les informations de comparaison.

Le tHDFSCCompare permet de contrôler la qualité des données traitées.

Propriétés du tHDFSCCompare Standard

Ces propriétés sont utilisées pour configurer le tHDFSCCompare s'exécutant dans le framework de Jobs Standard.

Le composant tHDFSCCompare Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Property type | Peut être Built-in ou Repository.  
Built-in : Propriétés utilisées ponctuellement.  
Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.  
Use an existing connection | Cochez cette case et, dans la liste Component List, cliquez sur le composant HDFS de connexion duquel vous souhaitez réutiliser les informations de connexion précédemment définies.  
Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.  
Distribution | Sélectionnez dans la liste le cluster que vous utilisez.  
Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
• Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com). |
• L’option **Custom** vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste [Hadoop configuration](en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend. Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez [Connexion à une distribution Hadoop personnalisée](à la page 1677).

<table>
<thead>
<tr>
<th>Hadoop version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use kerberos authentication</td>
<td>Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le &quot;principal name&quot; de Kerberos pour le NameNode</td>
</tr>
</tbody>
</table>
Dans le champ affiché, cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

  Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

| Use a keytab to authenticate | Cochez la case Use a keytab to authenticate pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

  Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser.

| NameNode URI | Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée masternode comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

| User name | Le champ User name est disponible lorsque vous n’utilisez pas Kerberos pour vous authentifier. Dans ce champ, saisissez votre identifiant pour cette distribution. Si vous laissez le champ vide, le nom de la machine hébergeant le Studio sera utilisé.

| Group | Identifiant de l’utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ
peut être disponible ou indisponible selon la distribution que vous utilisez.

### Schema et Edit schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur **Edit schema** afin de visualiser le schéma.

### Comparison mode
Sélectionnez le mode de comparaison à appliquer.

### File to compare
Parcourez votre système ou saisissez le chemin d’accès au fichier de HDFS dont vous souhaitez contrôler la qualité.

### Reference file
Parcourez votre système ou saisissez le chemin d’accès au fichier de HDFS sur lequel se base la comparaison.

### If differences detected, display et If no differences detected, display
Saisissez un message à afficher dans la console de la vue **Run**, selon les résultats de la comparaison.

### Print to console
Cochez cette case pour afficher le message dans la console de la vue **Run**.

---

### Advanced settings

#### Encoding
Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.

#### Hadoop properties
Le **Studio Talend** utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :  


### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

### Variables globales

| Variables globales | DIFFERENCE : résultat de la comparaison. Cette variable est une variable Flow et retourne un booléen.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | Le tHDFSCmpare peut être utilisé en standalone ou envoyer au composant suivant les informations générées.

| Dynamic settings | Cliquez sur le bouton [] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre
dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

| Prérequis | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

  Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

  Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

| Limitation | La version 1.6+ de JRE est requise pour exécuter le composant.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
### tHDFSCConnection

Ce composant se connecte à un HDFS donné afin que d’autres composants Hadoop puissent réutiliser la connexion créée pour communiquer avec HDFS.

Le tHDFSCConnection fournit une connexion au système de fichiers distribués Hadoop (HDFS) lors de l’exécution.

### Propriétés du tHDFSCConnection Standard

Ces propriétés sont utilisées pour configurer le tHDFSCConnection s’exécutant dans le framework de Jobs Standard.

Le composant tHDFSCConnection Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

#### Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Property type** | Peut être **Built-in** ou **Repository**.  
**Built-in** : Propriétés utilisées ponctuellement.  
**Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |
| **Distribution** | Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
- **Si vous sélectionnez Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).  
- **L’option Custom** vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.  
1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou, |
2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend, Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

<table>
<thead>
<tr>
<th><strong>Hadoop version</strong></th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inspect the classpath for configurations</strong></td>
<td>Cochez cette case pour permettre au composant de vérifier les fichiers de configuration dans le répertoire configuré pour la variable <code>$HADOOP_CONF_DIR</code> et de lire directement les paramètres de ces fichiers dans le répertoire. Cette fonctionnalité vous permet de modifier facilement la configuration Hadoop afin que le composant puisse passer d’un environnement à un autre, comme par exemple pour passer d’un environnement test à un environnement production.</td>
</tr>
</tbody>
</table>
Dans ce cas, les champs ou les options utilisée pour configurer la connexion Hadoop et/ou la sécurité Kerberos sont masqués.

Si vous souhaitez utiliser certains paramètres comme les paramètres Kerberos mais que ces paramètres ne sont pas inclus dans les fichiers de configuration Hadoop, vous devez créer un fichier appelé *talend-site.xml* et mettre ce fichier dans le répertoire défini dans `$HADOOP_CONF_DIR`. Le fichier *talend-site.xml* doit se présenter comme suit :

```xml
<!-- Put site-specific property overrides in this file. -->
<configuration>
    <property>
        <name>talend.kerberos.authentication</name>
        <value>kinit</value>
        <description>Set the Kerberos authentication method to use. Valid values are: kinit or keytab.</description>
    </property>
    <property>
        <name>talend.kerberos.keytab.principal</name>
        <value>user@BIGDATA.COM</value>
        <description>Set the keytab's principal name.</description>
    </property>
    <property>
        <name>talend.kerberos.keytab.path</name>
        <value>/kdc/user.keytab</value>
        <description>Set the keytab's path.</description>
    </property>
    <property>
        <name>talend.encryption</name>
        <value>none</value>
        <description>Set the encryption method to use. Valid values are: none or ssl.</description>
    </property>
    <property>
        <name>ssl.trustStore.path</name>
        <value>ssl</value>
        <description>Set SSL trust store path.</description>
    </property>
    <property>
        <name>ssl.trustStore.password</name>
        <value>ssl</value>
    </property>
</configuration>
```
Les paramètres lus depuis ces fichiers de configuration écrasent ceux utilisés par défaut dans le Studio. Lorsqu’un paramètre n’existe pas dans ces fichiers de configuration, le paramètre par défaut est utilisé.

### Use kerberos authentication

Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

### Use a keytab to authenticate

Cochez la case Use a keytab to authenticate pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser.

| NameNode URI | Saisissez l’URI du NameNode Hadoop. |
| User name | Nom d’utilisateur de HDFS. |
| Group | Identifiant de l’utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ |
peut être disponible ou indisponible selon la distribution que vous utilisez.

<table>
<thead>
<tr>
<th>Hadoop properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Le <em>Studio Talend</em> utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.</td>
</tr>
<tr>
<td>• Notez que, si vous utilisez les métadonnées stockées centralement dans le <strong>Repository</strong>, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste <strong>Property type</strong>, vous passiez de <strong>Repository</strong> à <strong>Built-in</strong>.</td>
</tr>
<tr>
<td>Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur <a href="http://hadoop.apache.org/docs">http://hadoop.apache.org/docs</a> en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use datanode hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez la case <strong>Use datanode hostname</strong> pour permettre au Job d’accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété <em>dfs.client.use.datanode.hostname</em> à true. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Setup HDFS encryption configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si le chiffrement transparent HDFS est activé dans votre cluster, cochez la case <strong>Setup HDFS encryption configurations</strong> et, dans le champ <strong>HDFS encryption key provider</strong> qui s’affiche, saisissez l’emplacement du proxy KMS.</td>
</tr>
<tr>
<td>Pour plus d’informations concernant le chiffrement transparent HDFS et son proxy KMS, consultez <a href="http://hadoop.apache.org/docs">Transparent Encryption in HDFS</a> (en anglais).</td>
</tr>
</tbody>
</table>
**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| Règle d’utilisation | Ce composant s’utilise avec d’autres composants Hadoop. |

| Prérequis | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les bibliothèques du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontrez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs |
en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

| Limitation       | La version 1.6+ de JRE est requise pour exécuter le composant. |

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tHDFSCopy

Ce composant copie un fichier ou dossier source vers un répertoire cible de HDFS et supprime la source si nécessaire.

Propriétés du tHDFSCopy Standard

Ces propriétés sont utilisées pour configurer le tHDFSCopy s’exécutant dans le framework de Jobs Standard.

Le composant tHDFSCopy Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Property type | Peut être Built-in ou Repository.  
|               | **Built-in** : Propriétés utilisées ponctuellement.  
|               | **Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.  |

| Use an existing connection | Cochez cette case et, dans la liste Component List, cliquez sur le composant HDFS de connexion duquel vous souhaitez réutiliser les informations de connexion précédemment définies.  
|                           | Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.  |

| Distribution | Sélectionnez dans la liste le cluster que vous utiliserez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
|             | • Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).  
|             | • L’option Custom vous permet de vous connecter à un cluster différente des clusters de la liste,  |
par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution.

Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use kerberos authentication</td>
<td>Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le “principal name” de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre...</td>
</tr>
</tbody>
</table>
identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

### Use a keytab to authenticate

Cochez la case Use a keytab to authenticate pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser.

### NameNode URI

Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée masternode comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

### User name

Le champ User name est disponible lorsque vous n’utilisez pas Kerberos pour vous authentifier. Dans ce champ, saisissez votre identifiant pour cette distribution. Si vous laissez le champ vide, le nom de la machine hébergeant le Studio sera utilisé.

### Group

Identifiant de l’utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ peut être disponible ou indisponible selon la distribution que vous utilisez.
### Source file or directory

Parcourez votre système ou saisissez le chemin d’accès aux données à utiliser dans le système de fichiers.

### Target location

Saisissez le chemin d’accès au répertoire de HDFS dans lequel vous souhaitez copier les données.

### Rename

Pour renommer le fichier ou dossier copié vers l’emplacement, cochez cette case afin d’afficher le champ **New name**. Saisissez le nouveau nom.

### Copy merge

Cochez cette case pour fusionner les fichiers *part* générés à la fin d’un calcul MapReduce. Une fois cochée, saisissez dans le champ **Merge name** le nom du fichier final fusionné.

### Remove source

Cochez cette case pour supprimer le fichier ou dossier source une fois la source copiée à l’emplacement cible.

### Override target file (This option does not override the directory)

Cochez cette case pour écraser le fichier déjà existant à l’emplacement cible. Cette option n’écrase pas le dossier.

### Advanced settings

#### Hadoop properties

Le *Studio Talend* utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste **Property type**, vous passiez de Repository à Built-in.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :

### Variables globales

<table>
<thead>
<tr>
<th>Variables globales</th>
<th>DESTINATION_FILEPATH : chemin d'accès au fichier cible. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SOURCE_FILEPATH : chemin d'accès au fichier source. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Le thDFSCopy peut être utilisé en standalone.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant.</td>
</tr>
<tr>
<td></td>
<td>La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu'un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.</td>
</tr>
<tr>
<td></td>
<td>Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à</td>
</tr>
</tbody>
</table>
### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d'exemple relatives à MapR.

- Assurez-vous d'avoir installé le client MapR sur la même machine que le Studio Talend et d'avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D'après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées `MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native`. Par exemple, pour Windows, la bibliothèque est `lib\MapRClient.dll` dans le fichier `jar` du client MapR. Pour plus d'informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

Si vous n'ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : `no MapRClient in java.library.path`.

- Configurez l'argument `-Djava.library.path`, par exemple, dans la zone *Job Run VM arguments* de la vue *Run/Debug* de la boîte de dialogue *[Preferences]* dans le menu *Window*. Cet argument fournit au studio le chemin d'accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d'utiliser entièrement l'aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d'informations concernant l'installation d'une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

### Limitation

La version 1.6+ de JRE est requise pour exécuter le composant.

### Scénario associé

Pour un scénario associé, consultez :

- *Procédure* à la page 1041.
- *Scénario : Effectuer une boucle sur un répertoire HDFS* à la page 1616.
tHDFSDelete

Ce composant supprime un fichier situé dans un système de fichiers distribué Hadoop (HDFS).

**Propriétés du tHDFSDelete Standard**

Ces propriétés sont utilisées pour configurer le tHDFSDelete s'exécutant dans le framework de Jobs Standard.

Le composant tHDFSDelete Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

### Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Property type** | Peut être *Built-in* ou *Repository.*  
  *Built-in*: Propriétés utilisées ponctuellement.  
  *Repository*: Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées. |
| **Use an existing connection** | Cochez cette case et, dans la liste Component List, cliquez sur le composant HDFS de connexion duquel vous souhaitez réutiliser les informations de connexion précédemment définies.  
Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
| **Distribution** | Sélectionnez dans la liste le cluster que vous utilisez.  
Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
- Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).  
- L’option Custom vous permet de vous connecter à un cluster différente des clusters de la liste, par exemple une distribution non supportée officiellement par Talend. |
1. Sélectionner *Import from existing version* pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner *Import from zip* pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans *Talend Exchange*, des membres de la Communauté *Talend* ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste *Hadoop configuration* (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option *Import from existing version*, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par *Talend*. *Talend* et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez *Connexion à une distribution Hadoop personnalisée* à la page 1677.

<table>
<thead>
<tr>
<th><strong>Hadoop version</strong></th>
<th>Sélectionnez la version de la distribution Hadoop que vous utiliserez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use kerberos authentication</strong></td>
<td>Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le &quot;principal name&quot; de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.</td>
</tr>
</tbody>
</table>
• Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

| Use a keytab to authenticate | Cochez la case Use a keytab to authenticate pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser. |
| --- | --- |

<table>
<thead>
<tr>
<th>NameNode URI</th>
<th>Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée masternode comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>User name</th>
<th>Nom d’utilisateur de HDFS.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Group</th>
<th>Identifiant de l’utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ peut être disponible ou indisponible selon la distribution que vous utilisez.</th>
</tr>
</thead>
</table>

| File or Directory Path | Renseignez ce champ avec le chemin d’accès au fichier ou au dossier à supprimer de HDFS. |
## Advanced settings

| Hadoop properties | Si vous devez utiliser la configuration personnalisée pour votre Hadoop, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) vont écraser celles définies précédemment pour Hadoop.

Pour plus d’informations concernant les propriétés requises par Hadoop, consultez la documentation Hadoop. |
|-------------------|--------------------------------------------------------------------------------------------------|

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

## Variables globales

| Variables globales | **DELETE_PATH** : chemin du dossier ou fichier supprimé. Cette variable est une variable *After* et retourne une chaine de caractères.

**CURRENT_STATUS** : résultat d’exécution du composant. Cette variable est une variable *After* et retourne une chaine de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaine de caractères.

Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en standalone.</th>
</tr>
</thead>
</table>

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant. |
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.


| Prérequis | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue [Preferences] dans le menu **Window**. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

| Limitation | La version 1.6+ de JRE est requise pour exécuter le composant.

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tHDFSEExist**

Ce composant vérifie l’existence d’un fichier dans un répertoire donné dans HDFS.

**Propriétés du tHDFSEExist Standard**

Ces propriétés sont utilisées pour configurer le tHDFSEExist s’exécutant dans le framework de Jobs Standard.

Le composant tHDFSEExist Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Property type | Peut être Built-in ou Repository.  
|               | **Built-in** : Propriétés utilisées ponctuellement.  
|               | **Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.  

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  

| Distribution | Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
|             | • Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).  
|             | • L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.  
|             | 1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée
et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
</table>
| Use kerberos authentication | Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.  
  • Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou |
comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

| Use a keytab to authenticate | Cochez la case Use a keytab to authenticate pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser. |
| NameNode URI | Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée masternode comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé. |
| User name | Le champ User name est disponible lorsque vous n’utilisez pas Kerberos pour vous authentifier. Dans ce champ, saisissez votre identifiant pour cette distribution. Si vous laissez le champ vide, le nom de la machine hébergeant le Studio sera utilisé. |
| Group | Identifiant de l’utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ peut être disponible ou indisponible selon la distribution que vous utilisez. |
| HDFS directory | Parcourez votre système ou saisissez le chemin d’accès aux données à utiliser dans le système de fichiers. |
| File name or relative path | Saisissez le nom du fichier dont vous souhaitez vérifier l’existence. Si nécessaire, parcourez votre système |
jusqu’au fichier ou saisissez le chemin d’accès relatif au répertoire défini dans le champ **HDFS directory**.

### Advanced settings

**Hadoop properties**

Le **Studio Talend** utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :


**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Variables globales

**Variables globales**

<table>
<thead>
<tr>
<th>Variables globales</th>
<th>EXIST : résultat spécifiant si un fichier existe ou non. Cette variable est une variable Flow et retourne un booléen.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>FILENAME</strong> : nom du fichier traité. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>
Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Le <strong>thDFSExit</strong> peut être utilisé en standalone.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
| **Prérequis** | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.  

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est *lib\MapRClient.dll* dans le fichier *jar* du client MapR. Pour plus d’informations, consultez la page |

### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est *lib\MapRClient.dll* dans le fichier *jar* du client MapR. Pour plus d’informations, consultez la page
Scénario : Vérifier l’existence d’un fichier dans HDFS

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.
Dans ce scénario, un Job à deux composants vérifie si un fichier spécifique existe dans HDFS et retourne un message pour indiquer le résultat de la vérification.
Dans un cas d’utilisation réelle, vous pouvez ensuite traiter le fichier en question selon le résultat de la vérification, à l’aide d’autres composants HDFS fournis avec le Studio Talend.

Lancez la distribution d’Hadoop dans laquelle se trouve le fichier dont vous souhaitez vérifier l’existence.

Relier les composants

Procédure
1. Dans la perspective Integration du Studio Talend, créez un Job vide, nommé hdfsexist_file, par exemple, à partir du nœud Job Designs de la vue Repository.
Pour plus d’informations concernant la création d’un Job, consultez le Guide utilisateur du Studio Talend.
2. Déposez un tHDFSExist et un tMsgBox dans l’espace de modélisation graphique.
3. Connectez-les à l’aide d’un lien Trigger > Run if.
Configurer la connexion à HDFS

Procédure

1. Double-cliquez sur le composant **tHDFSExist** pour ouvrir sa vue **Component**.

![tHDFSExist_1](image)

2. Dans la zone **Version**, sélectionnez la distribution d’Hadoop à laquelle vous vous connectez ainsi que sa version.

3. Dans la zone **Connection**, saisissez les valeurs des paramètres nécessaires à la connexion à HDFS. Dans un cas d’utilisation réelle, vous pouvez utiliser un **tHDFSConnection** pour créer une connexion et la réutiliser. Pour plus d’informations, consultez **tHDFSConnection** à la page 1561.


5. Dans le champ **File name or relative path**, saisissez le nom du fichier dont vous souhaitez vérifier l’existence. Par exemple, **output.csv**.

Configurer le message à retourner

Procédure

1. Double-cliquez sur le **tMsgBox** pour ouvrir sa vue **Component**.

![tMsgBox_1](image)

1. Dans la zone **Title**, saisissez le titre que vous voulez que le message affiche.

2. Dans la zone **Buttons**, choisissez le bouton de confirmation qui apparaîtra.

3. Dans la zone **Icon**, choisissez un pictogramme d'erreur pour le message.

4. Dans la zone **Message**, saisissez le message que vous voulez afficher.

"This file does not exist!"
2. Dans le champ **Title**, saisissez le titre à utiliser dans la fenêtre de message en cours de création.
3. Dans la liste **Buttons**, sélectionnez **OK** pour choisir le bouton affiché dans la fenêtre de message.
4. Dans la liste **Icon**, sélectionnez **Icon information**.
5. Dans le champ **Message**, saisissez le message que vous souhaitez afficher une fois la vérification effectuée. Dans cet exemple, saisissez "This file does not exist!".

### Configurer la condition

**Procédure**

1. Cliquez sur le lien **If** pour ouvrir sa vue **Basic settings**, dans laquelle vous pouvez configurer la condition pour vérifier l’existence du fichier.

![Image d'interface de configuration](image)

2. Dans le champ **Condition**, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables et sélectionnez la variable globale **EXISTS**. Saisissez un point d'exclamation avant la variable afin de mettre la variable à la forme négative.

### Exécuter le Job

**Procédure**

Appuyez sur **F6** pour exécuter le Job.

**Résultats**

Une fois exécuté, une fenêtre de message s'ouvre et indique que le fichier nommé output.csv n'existe pas dans le répertoire défini précédemment.
Dans la distribution de HDFS dans laquelle vous avez vérifié l’existence du fichier, parcourez votre système jusqu’au répertoire spécifié. Vous pouvez constater que le fichier n’existe pas.

**Contents of directory /user/ychen/data/hdfs/dest**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Replication</th>
<th>Block Size</th>
<th>Modification Time</th>
<th>Permission</th>
<th>Owner</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer_m.csv</td>
<td>file</td>
<td>1.02 KB</td>
<td>3</td>
<td>64 MB</td>
<td>2012-08-27 12:58</td>
<td>rw-r--r--</td>
<td>ychen</td>
<td>hadoop</td>
</tr>
<tr>
<td>customer_s.csv</td>
<td>file</td>
<td>1.11 KB</td>
<td>3</td>
<td>64 MB</td>
<td>2012-08-27 12:58</td>
<td>rw-r--r--</td>
<td>ychen</td>
<td>hadoop</td>
</tr>
<tr>
<td>out.csv</td>
<td>file</td>
<td>0.99 KB</td>
<td>3</td>
<td>64 MB</td>
<td>2012-08-27 12:42</td>
<td>rw-r--r--</td>
<td>ychen</td>
<td>hadoop</td>
</tr>
<tr>
<td>out2.csv</td>
<td>file</td>
<td>1.11 KB</td>
<td>3</td>
<td>64 MB</td>
<td>2012-08-27 12:42</td>
<td>rw-r--r--</td>
<td>ychen</td>
<td>hadoop</td>
</tr>
</tbody>
</table>

Go back to DFS home
**tHDFSGet**

Ce composant copie des fichiers d'un système de fichiers distribué Hadoop (HDFS) et les colle dans un répertoire défini par l'utilisateur. Si nécessaire, il renomme ces fichiers.

Le tHDFSGet se connecte au système de fichiers distribués Hadoop, permet d'obtenir de gros fichiers avec une performance optimisée.

**Propriétés du tHDFSGet Standard**

Ces propriétés sont utilisées pour configurer le tHDFSGet s'exécutant dans le framework de Jobs Standard.

Le composant tHDFSGet Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
</tbody>
</table>

| Use an existing connection   | Cochez cette case et, dans la liste Component List, cliquez sur le composant HDFS de connexion duquel vous souhaitez réutiliser les informations de connexion précédemment définies.  |
|-------------------------------| Notez que lorsqu'un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.  |

| Distribution                  | Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  |
|                               | • Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).  |
• L’option **Custom** vous permet de vous connecter à un cluster différent de celui de la liste, par exemple une distribution non supportée officiellement par **Talend**.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste [Hadoop configuration](en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les Jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend**. **Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez [Connexion à une distribution Hadoop personnalisée](en anglais) à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use kerberos authentication</td>
<td>Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le &quot;principal name&quot; de Kerberos pour le NameNode</td>
</tr>
</tbody>
</table>
dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

### Use a keytab to authenticate

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser.

### NameNode URI

Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée *masternode* comme NameNode, son emplacement est *hdfs://masternode:portnumber*. Si vous utilisez WebHDFS, l’emplacement doit être *webhdfs://masternode:portnumber*. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être *swebhdfs* et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

### User name

Le champ **User name** est disponible lorsque vous n’utilisez pas Kerberos pour vous authentifier. Dans ce champ, saisissez votre identifiant pour cette distribution. Si vous laissez le champ vide, le nom de la machine hébergeant le Studio sera utilisé.

### Group

Identifiant de l’utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ
peut être disponible ou indisponible selon la distribution que vous utilisez.

<table>
<thead>
<tr>
<th><strong>HDFS directory</strong></th>
<th>Parcourez votre système ou saisissez le chemin d'accès aux données à utiliser dans le système de fichiers.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Local directory</strong></td>
<td>Dossier local où stocker les fichiers obtenus via HDFS.</td>
</tr>
<tr>
<td><strong>Overwrite file</strong></td>
<td>Option permettant d'écraser ou non les fichiers avec le nouveau fichier.</td>
</tr>
<tr>
<td><strong>Append</strong></td>
<td>Cochez cette case pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
<tr>
<td><strong>Include subdirectories</strong></td>
<td>Cochez cette case si la source d'entrée comprend des sous-répertoires.</td>
</tr>
</tbody>
</table>

**Files**

Dans la zone Files, les champs à remplir sont :
- **File mask** : saisissez le nom du fichier qui sera sélectionné dans HDFS. Vous pouvez utiliser des expressions régulières.
- **New name** : renommez le fichier obtenu.

| **Die on error** | Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. |

### Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Hadoop properties**

Le Studio Talend utilise une configuration par défaut pour son moteur, afin d'effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste Property type, vous passiez de Repository à Built-in.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d'Apache Hadoop sur http://hadoop.apache.org/docs en sélectionnant la version de la documentation souhaitée. À titre d'exemple, les liens vers certaines propriétés sont listés ci-après :
• Généralement, les propriétés relatives à HDFS peuvent être trouvées dans le fichier hdfs-default.xml correspondant à votre distribution, comme par exemple http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais).

• Apache fournit également une page listant toutes les propriétés relatives à Hive : http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais).

**Variables globales**

<table>
<thead>
<tr>
<th>Variables globales</th>
<th>NB_FILE : Indique le nombre de fichiers traités. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CURRENT_STATUS : résultats d'exécution du composant. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>TRANSFER_MESSAGES : informations transférées du fichier. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant combine la connexion HDFS et l’extraction de données. Il est utilisé en standalone comme sous-job pour déplacer des données de HDFS vers un répertoire local donné.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A la différence des composants tHDFSInput et tHDFSOutput, il s’exécute en standalone et ne génère pas de flux d’entrée ou de sortie pour les autres composants.</td>
</tr>
<tr>
<td></td>
<td>Il est souvent connecté au Job utilisant les liens OnSubjobOk ou OnComponentOk, en fonction du contexte.</td>
</tr>
</tbody>
</table>

| Dynamic settings    | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de |
contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu'un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend.

La liste suivante présente des informations d’exemple relatives à MapR.

- **Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine.** D’après la documentation de MapR, la ou les bibliothèques du client MapR correspondant à chaque OS peuvent être trouvées `MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native`. Par exemple, pour Windows, la bibliothèque est `lib\MapRClient.dll` dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

> Si vous n’ajoutez pas de librairie, il est possible que vous rencontrez l’erreur suivante : `no MapRClient in java.library.path`.

- **Configurez l’argument** `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue [Preferences] dans le menu **Window**. Cet argument fournit au studio le chemin d'accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d'utiliser entièrement l'aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.
Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

| Limitation | La version 1.6+ de JRE est requise pour exécuter le composant. |

Scénario : Traiter des données avec le système de fichiers distribués Hadoop

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Le scénario suivant décrit un simple Job qui crée un fichier dans un dossier défini, dans ou en dehors d’HDFS, l’enregistre dans un autre dossier local et le lit à la fin de l’exécution du Job.

Construire le Job

Procédure

1. Glissez les composants suivants de la Palette vers l’espace de modélisation : tFixedFlowInput, tFileOutputDelimited, tHDFSPut, tHDFSGet, tFileInputDelimited et tLogRow.
2. Connectez le tFixedFlowInput au tFileOutputDelimited à l’aide d’un lien Row > Main.
3. De la même façon, cliquez-droit sur le composant tFileInputDelimited pour le relier au tLogRow, en utilisant le lien Row Main.
5. Reliez le tHDFSPut au tHDFSGet à l’aide d’un lien OnSubjobOk.
6. Reliez le composant tHDFSGet au tFileInputDelimited à l’aide d’un lien OnSubjobOk.
Configurer le composant d’entrée

Procédure

1. Dans l’espace de modélisation, double-cliquez sur le composant **tFixedFlowInput** pour définir ses **Basic settings**.

2. Dans la liste **Schema**, sélectionnez **Built-In** et cliquez sur le bouton [...] situé à côté d’**Edit Schema** afin d’afficher la structure des données que vous souhaitez créer à partir des variables internes. Dans ce scénario, le schéma contient une colonne : **content**.

3. Cliquez sur le bouton [+ ] pour ajouter une ligne de paramètres.

4. Cliquez sur **OK** pour fermer la boîte de dialogue et accepter la propagation des modifications.
5. Dans l’onglet **Basic settings**, dans la zone **Mode**, définissez la valeur de la colonne, en utilisant l’option **Use Single Table**. Dans ce scénario, la valeur est "Hello world!".

### Configurer le composant tFileOutputDelimited

**Procédure**

1. Dans l’espace de modélisation, sélectionnez le composant **tFileOutputDelimited** et cliquez sur l’onglet **Component** afin de définir sa configuration de base (**Basic settings**).

2. Cliquez sur le bouton [...] situé à côté du champ **File Name** et parcourez votre répertoire jusqu’au fichier de sortie dans lequel vous souhaitez écrire des données, *in.txt* dans cet exemple.

### Charger les données depuis le fichier local

**Procédure**

1. Dans l’espace de modélisation, sélectionnez le composant **tHDFSPut** et cliquez sur l’onglet **Component** pour définir ses **Basic settings**.
2. Sélectionnez Apache 0.20.2 dans la liste Hadoop version.

3. Dans les champs Host, Port, Username et Group, saisissez les informations de connexion à HDFS. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

4. À côté du champ Local directory, cliquez sur le bouton [...] afin de parcourir votre répertoire et sélectionner le fichier à charger dans HDFS. Dans ce scénario, le dossier a été spécifié pendant la configuration du composant tFileOutputDelimited : C:/hadoopfiles/putFile/.

5. Dans le champ HDFS directory, saisissez l’emplacement défini dans HDFS pour enregistrer le fichier à charger. Dans cet exemple, il s’agit de /testFile.

6. Cliquez sur le champ Overwrite file pour développer le menu déroulant.

7. Dans le menu, sélectionnez always (toujours).

8. Dans la zone Files, cliquez sur le bouton [+] pour ajouter une ligne dans laquelle vous définissez le fichier à charger.


**Obtenir les données de HDFS**

**Procédure**

1. Dans l’espace de modélisation, sélectionnez le composant tHDFSGet et cliquez sur l’onglet Component pour définir sa configuration de base.
2. Sélectionnez **Apache 0.20.2** dans la liste **Hadoop version**.

3. Dans les champs **NameNode URI, Username, Group**, saisissez les paramètres de connexion à HDFS. Si vous utilisez WebHDFS, l'emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL, le schéma d'URI doit être `swebhdfs` et vous devez utiliser un `tLibraryLoad` dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

4. Dans le champ **HDFS directory**, saisissez l'emplacement de stockage du fichier chargé dans HDFS. Dans cet exemple, il s'agit de `/testFile`.

5. A côté du champ **Local directory**, cliquez sur le bouton ` [...]` pour parcourir jusqu'au dossier spécifié pour l'enregistrement des fichiers extraits d'HDFS. Dans ce scénario, le dossier est : `C:/hadoopfiles/getFile/`.

6. Cliquez sur le champ **Overwrite file** pour développer le menu déroulant.

7. Dans le menu, sélectionnez **always** (toujours).

8. Dans la zone **Files**, cliquez sur le bouton `[+]` pour ajouter une ligne dans laquelle vous définissez le fichier à extraire.


**Lire des données de HDFS et les sauvegarder localement**

**Procédure**

1. Dans l'espace de modélisation, sélectionnez le composant **tFileInputDelimited**. Cliquez sur l'onglet **Component** pour définir ses **Basic settings**.
2. Dans la liste **Property Type**, sélectionnez le type **Built-In**.

3. A côté du champ **File Name/Stream**, cliquez sur le bouton [...] pour parcourir votre répertoire jusqu'au fichier obtenu de HDFS. Dans ce scénario, le fichier est C:/hadoopfiles/getFile/in.txt.

4. Dans la liste **Schema**, sélectionnez **Built-In** et cliquez sur **Edit schema** pour définir les données à passer au composant **tLogRow**.

5. Cliquez sur le bouton [+] pour ajouter une nouvelle colonne.

6. Cliquez sur **OK** pour fermer la boîte de dialogue et accepter la propagation des modifications.

**Exécuter le Job**

Sauvegardez le Job et appuyez sur la touche F6 pour l’exécuter.

Le fichier **in.txt** est créé et chargé dans HDFS.
Le fichier est aussi extrait de HDFS à l'aide du composant 
**tHDFSGet** et lu par le composant 
**tFileInputDelimited**.

---

Starting job feature11712_hadoop_get_put at 18:09 04/05/2010.

[statistics] connecting to socket on port 3637
[statistics] connected
hello world!
[statistics] disconnected
Job feature11712_hadoop_get_put ended at 18:09 04/05/2010.
exit code 01
tHDFSInput

Ce composant extrait les données d'un fichier HDFS afin que d'autres composants puissent les traiter. Le tHDFSInput lit un fichier situé dans un système de fichiers distribués Hadoop (HDFS) et écrit les données qui vous intéressent à partir de ce fichier dans un schéma Talend. Il passe ensuite les données au composant suivant.

Propriétés du tHDFSInput Standard

Ces propriétés sont utilisées pour configurer le tHDFSInput s’exécutant dans le framework de Jobs Standard.

Le composant tHDFSInput Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être <strong>Built-In</strong> ou <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-In</strong> : propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <em>line</em> lors du nommage des champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cliquez sur <a href="#">Edit schema</a> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <a href="#">View schema</a> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <a href="#">Change to built-in property</a> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <a href="#">Update repository connection</a> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <a href="#">No</a> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td></td>
<td><strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <a href="#">Guide utilisateur du Studio Talend</a>.</td>
</tr>
</tbody>
</table>

### Use an existing connection

Cochez cette case et, dans la liste Component List, cliquez sur le composant HDFS de connexion duquel vous souhaitez réutiliser les informations de connexion précédemment définies.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.

### Distribution

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez. Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center ([https://help.talend.com](https://help.talend.com)).

- L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste [Hadoop configuration](https://help.talend.com) (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une
Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’im porter les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
</table>

**Use kerberos authentication**

Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le “principal name” de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

  Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication et Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

  Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

| Use a keytab to authenticate | Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et les |
chemin d'accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser.

### NameNode URI
Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée masternode comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l'emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

### User name
Le champ **User name** est disponible lorsque vous n’utilisez pas Kerberos pour vous authentifier. Dans ce champ, saisissez votre identifiant pour cette distribution. Si vous laissez le champ vide, le nom de la machine hébergeant le Studio sera utilisé.

### Group
Identifiant de l’utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ peut être disponible ou indisponible selon la distribution que vous utilisez.

### File Name
Parcourez votre système ou saisissez le chemin d’accès aux données à utiliser dans le système de fichiers.

Si le chemin que vous avez saisi pointe vers un dossier, le composant lit tous les fichiers contenus dans ce dossier. De plus, si le dossier contient des sous-dossiers et que vous souhaitez que les fichiers de ces sous-dossiers soient lus, cochez la case **Include sub-directories if path is directory** dans la vue **Advanced settings**.

### Type
Sélectionnez le type de fichier à traiter. Le type de fichier peut être :

- **Text file**.

Une fois le format **Sequence file** sélectionné, les listes **Key column** et **Value column** apparaissent et vous permettent de sélectionner les clés et les valeurs de ce fichier de séquence à traiter.
### THDFSInput

<table>
<thead>
<tr>
<th>Paramètre</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row separator</strong></td>
<td>Saisissez le séparateur utilisé pour identifier la fin d’une ligne. Ce champ n’est pas disponible pour un fichier Sequence.</td>
</tr>
<tr>
<td><strong>Field separator</strong></td>
<td>Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées. Ce champ n’est pas disponible pour un fichier Sequence.</td>
</tr>
<tr>
<td><strong>Header</strong></td>
<td>Saisissez le nombre de lignes d’en-tête à ignorer dans les données transférées. Par exemple, saisissez 0 pour ne pas ignorer de ligne dans les données sans en-tête et 1 pour les données dont l’en-tête se trouve dans la première ligne. Ce champ n’est pas disponible pour un fichier Sequence.</td>
</tr>
<tr>
<td><strong>Custom encoding</strong></td>
<td>Il est possible de rencontrer des problèmes d’encodage lorsque vous traitez les données stockées. Dans ce cas, cochez cette case pour afficher la liste Encoding. Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données. Ce champ n’est pas disponible pour un fichier Sequence.</td>
</tr>
<tr>
<td><strong>Compression</strong></td>
<td>Cochez la case Uncompress the data pour décompresser les données d’entrée. Hadoop fournit différents formats de compression permettant de réduire l’espace nécessaire au stockage des fichiers et d’accélérer le transfert de données. Lorsque vous lisez un fichier compressé, le Studio Talend doit le décompresser avant de pouvoir en alimenter le flux d’entrée. Ce champ n’est pas disponible pour un fichier Sequence.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Paramètre</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Include sub-directories if path is directory</strong></td>
<td>Cochez cette case pour lire non seulement le dossier spécifié dans le champ File name mais également les sous-dossiers contenus dans ce dossier.</td>
</tr>
<tr>
<td><strong>Hadoop properties</strong></td>
<td>Le Studio Talend utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.</td>
</tr>
</tbody>
</table>

- Notez que, si vous utilisez les métdonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métdonnées et passe en lecture seule jusqu’à ce
que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur http://hadoop.apache.org/docs en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :

- Généralement, les propriétés relatives à HDFS peuvent être trouvées dans le fichier `hdfs-default.xml` correspondant à votre distribution, comme par exemple http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais).

### Variables globales

**Variables globales**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

**Règle d’utilisation**

Ce composant requiert un lien de sortie.

**Dynamic settings**

Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans
différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Prérequis**

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib
\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

  Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

**Limitation**

La version 1.6+ de JRE est requise pour exécuter le composant.
Scénario associé

Pour un scénario associé, consultez :

- **Scénario 1 : Écrire des données dans un fichier délimité** à la page 1172.
- **Scénario : Traiter des données avec le système de fichiers distribués Hadoop** à la page 1594.
tHDFSList

Ce composant récupère une liste de fichiers ou dossiers à partir d'un masque de fichier et effectue une boucle sur chaque unité.

**Propriétés du tHDFSList Standard**

Ces propriétés sont utilisées pour configurer le tHDFSList s'exécutant dans le framework de Jobs Standard.

Le composant tHDFSList Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Basic settings**

| Property type       | Peut être **Built-in** ou **Repository**.  
|---------------------|------------------------------------------|
|                     | **Built-in** : Propriétés utilisées ponctuellement.  
|                     | **Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées. |

**Use an existing connection**

Cochez cette case et, dans la liste **Component List**, cliquez sur le composant HDFS de connexion duquel vous souhaitez réutiliser les informations de connexion précédemment définies.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste **Component List** présente uniquement les composants de connexion du Job du même niveau.

**Distribution**

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.

- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).  
- L’option **Custom** vous permet de vous connecter à un cluster différente des clusters de la liste,
par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner *Import from existing version* pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner *Import from zip* pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option *Import from existing version*, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use kerberos authentication</td>
<td>Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le &quot;principal name&quot; de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre</td>
</tr>
</tbody>
</table>
identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

  Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

### Use a keytab to authenticate

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est **user1** et le principal à utiliser est **guest**. Dans cette situation, assurez-vous que **user1** a les droits de lecture pour le fichier keytab à utiliser.

### NameNode URI

Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée **masternode** comme NameNode, son emplacement est `hdfs://masternode:portnumber`. Si vous utilisez WebHDFS, l’emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être `swebhdfs` et vous devez utiliser un `TLibraryLoad` dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

### User name

Le champ **User name** est disponible lorsque vous n’utilisez pas Kerberos pour vous authentifier. Dans ce champ, saisissez votre identifiant pour cette distribution. Si vous laissez le champ vide, le nom de la machine hébergeant le Studio sera utilisé.

### Group

Identifiant de l’utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ peut être disponible ou indisponible selon la distribution que vous utilisez.
<table>
<thead>
<tr>
<th><strong>HDFS Directory</strong></th>
<th>Parcourez votre système ou saisissez le chemin d’accès aux données à utiliser dans le système de fichiers.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FileList Type</strong></td>
<td>Sélectionnez dans la liste le type d’entrée sur lequel iterer : File si l’entrée est un ensemble de fichiers, Directories si l’entrée est un ensemble de répertoires, Both si l’entrée est un mélange des deux types ci-dessus.</td>
</tr>
<tr>
<td><strong>Include subdirectories</strong></td>
<td>Cochez cette case si la source d’entrée comprend des sous-répertoires.</td>
</tr>
<tr>
<td><strong>Case Sensitive</strong></td>
<td>Sélectionnez dans la liste le mode de casse, afin de créer des filtres sur les noms de fichiers, sensibles à la casse ou non.</td>
</tr>
<tr>
<td><strong>Use Glob Expressions as Filemask</strong></td>
<td>Cette case est cochée par défaut. Elle permet de filtrer les résultats à l’aide d’une Expression Globale (Glob Expressions).</td>
</tr>
<tr>
<td><strong>Files</strong></td>
<td>Cliquez sur le bouton [+1] pour ajouter autant de lignes de filtre que nécessaire : Filemask : Dans les lignes de filtre ajoutées, saisissez un nom ou un masque de fichier à l’aide de caractères spéciaux ou d’expressions régulières.</td>
</tr>
</tbody>
</table>
| **Order by** | Les dossiers sont listés en premier, les fichiers ensuite. Vous pouvez choisir de personnaliser l’ordre des dossier et des fichiers : By default : par ordre alphabétique, par dossier puis fichier, By file name : par ordre alphabétique ou ordre alphabétique inversé, By file size : du plus petit au plus grand ou du plus grand au plus petit, By modified date : du plus récent au plus ancien ou du plus ancien au plus récent.  

**Remarque** : Si vous sélectionnez by file name et que des fichiers ont le même nom, l’option modified date prime. Si vous sélectionnez by file size, si vous possédez des fichiers de taille identique, l’option file name prime. Si vous sélectionnez modified date et que des fichiers ont été modifiés à la même date, l’option file name prime. |
| **Order action** | Sélectionnez l’ordre de tri en cliquant sur l’un des boutons radio : ASC : ordre ascendant. DESC : ordre descendant. |
## Advanced settings

| **Use Exclude Filemask** | Cochez cette case pour activer le champ *Exclude Filemask* et exclure les conditions de filtre selon le type de fichier :  

*Exclude Filemask* : Saisissez dans le champ les types de fichiers à exclure, du tableau *Filemasks* dans la vue *Basic settings*.  

**Remarque :** Les types de fichier doivent être entre guillemets doubles et séparés par une virgule. |
|---------------------------------------------------------------|
| **Hadoop properties** | Le *Studio Talend* utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.  

- Notez que, si vous utilisez les métadonnées stockées centralement dans le *Repository*, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste *Property type*, vous passiez de *Repository* à *Built-in*.  

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :  


| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

## Variables globales

<table>
<thead>
<tr>
<th>Variables globales</th>
<th><strong>CURRENT_FILE</strong> : nom du fichier courant. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</th>
</tr>
</thead>
</table>
**CURRENT_FILEDIRECTORY** : répertoire du fichier courant. Cette variable est une variable Flow et retourne une chaîne de caractères.

**CURRENT_FILEEXTENSION** : extension du fichier courant. Cette variable est une variable Flow et retourne une chaîne de caractères.

**CURRENT_FILEPATH** : nom du fichier courant ainsi que son chemin d’accès. Cette variable est une variable Flow et retourne une chaîne de caractères.

**NB_FILE** : nombre de fichiers itérés. Cette variable est une variable Flow et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Le composant tHDFSList fournit une liste de fichiers ou dossiers à partir du répertoire HDFS sur lequel il effectue une boucle.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant.  
La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.  
Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir |
de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

### Connections

Liens de sortie (de ce composant à un autre) :

**Row** : Iterate

**Trigger** : OnSubjobOk ; OnSubjobError ; Run if ; OnComponentOk ; OnComponentError.

Liens d’entrée (d’un autre composant à celui-ci) :

**Row** : Iterate.

**Trigger** : Run if ; OnSubjobOk ; OnSubjobError ; OnComponentOk ; OnComponentError ; Synchronize ; Parallelize.

Pour plus d’informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.

### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le *Studio Talend* et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées `MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native`. Par exemple, pour Windows, la bibliothèque est `lib\MapRClient.dll` dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

  Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]** dans le menu **Window**. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

### Limitation

La version 1.6+ de JRE est requise pour exécuter le composant.
**Scénario : Effectuer une boucle sur un répertoire HDFS**

Ce scénario s'applique uniquement aux solutions Talend avec Big Data.

Ce scénario présente un Job à deux composants itérant un répertoire spécifié dans HDFS afin de sélectionner les fichiers et les déplacer vers un répertoire local.

**Préparer les données à utiliser**

**Procédure**

Créez les fichiers à itérer dans votre HDFS. Dans ce scénario, deux fichiers sont créés dans le répertoire : `/user/ychen/data/hdfs/out`.

### Contents of directory `/user/ychen/data/hdfs/out`

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Replication</th>
<th>Block Size</th>
<th>Modification Time</th>
<th>Permission</th>
<th>Owner</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>out.csv</td>
<td>file</td>
<td>1.01 KB</td>
<td>3</td>
<td>64 MB</td>
<td>2012-08-08 18:09</td>
<td>rw-r--r--</td>
<td>ychen</td>
<td>hadoop</td>
</tr>
<tr>
<td>out2.csv</td>
<td>file</td>
<td>1.1 KB</td>
<td>3</td>
<td>64 MB</td>
<td>2012-08-08 18:09</td>
<td>rw-r--r--</td>
<td>ychen</td>
<td>hadoop</td>
</tr>
</tbody>
</table>

**Local logs**

Log directory

This is Apache Hadoop release 1.0.3

Vous pouvez créer un Job dans le Studio Talend pour créer deux fichiers. Pour plus d'informations, consultez `tHDFSPut` à la page 1642 ou `tHDFSOutput` à la page 1620.

**Relier les composants**

**Procédure**

1. Dans la perspective Integration du Studio Talend, créez un Job vide nommé `HDFSList`, par exemple, dans le nœud Job Designs de la vue Repository.
Pour plus d’informations concernant la création d’un Job, consultez le Guide utilisateur du Studio Talend.

2. Déposez un composant `tHDFSList` et un `tHDFSGet` dans l’espace de modélisation graphique.
3. Connectez-les à l’aide d’un lien `Row > Iterate`.

**Configurer l’itération**

**Procédure**

1. Double-cliquez sur le composant `tHDFSList` pour ouvrir sa vue `Component`.

2. Dans la zone **Version**, sélectionnez la distribution d’Hadoop à laquelle vous êtes connecté ainsi que sa version.

3. Dans la zone **Connection**, saisissez les valeurs des paramètres requis pour vous connecter à HDFS. Dans un cas d’utilisation réelle, vous pouvez utiliser le composant `tHDFSConnection` afin de créer une connexion et la réutiliser à partir du composant en question. Pour plus d’informations, consultez `tHDFSConnection` à la page 1561.

4. Dans le champ **HDFS Directory**, saisissez le chemin d’accès au dossier dans lequel se trouvent les fichiers à iterer. Dans cet exemple, le répertoire est `/user/ychen/data/hdfs/out/`.

5. Dans le champ **FileList Type**, sélectionnez **File**.

6. Dans la table **Files**, cliquez sur le bouton pour ajouter une ligne et saisissez `*` entre guillemets afin d’effectuer une boucle sur les fichiers existants.
Sélectionner les fichiers

Procédure

1. Double-cliquez sur le tHDFSGet pour ouvrir sa vue Component.

2. Dans la zone Version, sélectionnez la distribution d'Hadoop à laquelle vous vous connectez, ainsi que sa version.

3. Dans la zone Connection, saisissez les valeurs des paramètres requis pour vous connecter à HDFS. Dans un cas d'utilisation réelle, vous pouvez utiliser le composant tHDFSConnection afin de créer une connexion et la réutiliser à partir du composant en question. Pour plus d’informations, consultez tHDFSConnection à la page 1561.

4. Dans le champ HDFS directory, saisissez le chemin d'accès au dossier dans lequel se trouvent les fichiers. Vous pouvez passer votre curseur sur ce champ puis appuyer sur les touches Ctrl+Espace pour afficher la liste d'auto-complétion et sélectionner la variable tHDFSList_1_CURRENT_FILEDIRECTORY afin de réutiliser le répertoire défini dans le composant tHDFSList. Dans cette variable, tHDFSList_1 est le libellé du composant. Si vous le nommez différemment, sélectionnez la bonne variable.

Une fois la variable sélectionnée, le répertoire devient, par exemple ((String)globalMap.get("tHDFSList_1_CURRENT_FILEDIRECTORY")) dans ce champ.

Pour plus d’informations concernant le nommage des composants, consultez le Guide utilisateur du Studio Talend.

5. Dans le champ Local directory, saisissez le chemin d'accès ou parcourez votre système jusqu'au dossier dans lequel vous souhaitez placer les fichiers sélectionnés. Ce dossier sera créé s’il n’existe pas. Dans cet exemple, le dossier est C:/hdfsFiles.

6. Dans le champ Overwrite file, sélectionnez always.
7. Dans la table **Files**, cliquez sur le bouton pour additionner une ligne et saisissez `*` entre guillemets dans la colonne **Filemask**, afin de récupérer tous les fichiers existants.

**Exécuter le Job**

**Procédure**

Appuyez sur la touche **F6** pour exécuter ce Job.

**Résultats**

Une fois le Job terminé, vous pouvez vérifier que les fichiers ont été créés dans le répertoire local.
tHDFSOutput

Ce composant écrit les flux de données qu’il reçoit dans un système de fichiers distribués Hadoop donné (HDFS).

Propriétés du tHDFSOutput Standard

Ces propriétés sont utilisées pour configurer le tHDFSOutput s’exécutant dans le framework de Jobs Standard.

Le composant tHDFSOutput Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Property type | Peut être Built-In ou Repository.  
|               | **Built-In** : propriétés utilisées ponctuellement.  
|               | **Repository** : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
|                       | Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|                       | - **View schema** : sélectionnez cette option afin de voir le schéma.  
|                       | - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
|                       | - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content.

| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |
| Use an existing connection | Cochez cette case et, dans la liste **Component List**, cliquez sur le composant HDFS de connexion duquel vous souhaitez réutiliser les informations de connexion précédemment définies.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste **Component List** présente uniquement les composants de connexion du Job du même niveau. |
| --- | --- |
| Distribution | Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option **Custom** vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend**. **Talend** et sa Communauté fournissent l’opportunité de vous
connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d'Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
</table>
| **Use kerberos authentication** | Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d'utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.  
- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.  
  Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.  
  Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez. |
| **Use a keytab to authenticate** | Cochez la case Use a keytab to authenticate pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend. |
Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est `user1` et le principal à utiliser est `guest`. Dans cette situation, assurez-vous que `user1` a les droits de lecture pour le fichier keytab à utiliser.

<table>
<thead>
<tr>
<th>NameNode URI</th>
<th>Saisissez l'URI du NameNode Hadoop, nœud maître d'un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée <code>masternode</code> comme NameNode, son emplacement est <code>hdfs://masternode:portnumber</code>. Si vous utilisez WebHDFS, l'emplacement doit être <code>webhdfs://masternode:portnumber</code>. Si ce WebHDFS est sécurisé via SSL, le schéma d'URI doit être <code>swebhdfs</code> et vous devez utiliser un <code>TLibraryLoad</code> dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.</th>
</tr>
</thead>
<tbody>
<tr>
<td>User name</td>
<td>Le champ <strong>User name</strong> est disponible lorsque vous n'utilisez pas Kerberos pour vous authentifier. Dans ce champ, saisissez votre identifiant pour cette distribution. Si vous laissez le champ vide, le nom de la machine hébergeant le Studio sera utilisé.</td>
</tr>
<tr>
<td>Group</td>
<td>Identifiant de l'utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ peut être disponible ou indisponible selon la distribution que vous utilisez.</td>
</tr>
<tr>
<td>File Name</td>
<td>Parcourez votre système ou saisissez le chemin d'accès au fichier de HDFS dans lequel vous avez écrit les données qui vous intéressent. Ce fichier est automatiquement créé s'il n'existe pas.</td>
</tr>
<tr>
<td>Type</td>
<td>Sélectionnez le type de fichier à traiter. Le type de fichier peut être :</td>
</tr>
<tr>
<td></td>
<td>• <strong>Text file</strong>.</td>
</tr>
</tbody>
</table>

Une fois le format **Sequence file** sélectionné, les listes **Key column** et **Value column** apparaissent et vous permettent de sélectionner les clés et les valeurs de ce fichier de séquence à traiter. |
| Action       | Sélectionnez une opération dans HDFS : |
|              | **Create** : Crée un fichier avec des données à l'aide du nom de fichier défini dans le champ **File Name**. |
|              | **Overwrite** : Écrase les données du fichier spécifié dans le champ **File Name**. |
**Append** : Insère les données dans le fichier spécifié dans le champ **File Name**. Le fichier spécifié est créé automatiquement s'il n'existe pas.

| Row separator | Saisissez le séparateur utilisé pour identifier la fin d'une ligne. Ce champ n'est pas disponible pour un fichier Sequence. |
| Field separator | Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées. Ce champ n'est pas disponible pour un fichier Sequence. |
| Custom encoding | Il est possible de rencontrer des problèmes d'encodage lorsque vous traitez les données stockées. Dans ce cas, cochez cette case pour afficher la liste **Encoding**. Sélectionnez l'encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données. Ce champ n'est pas disponible pour un fichier Sequence. |
| Compression | Cochez la case **Compress the data** afin de compresser les données de sortie. Hadoop fournit différents formats de compression permettant de réduire l'espace nécessaire au stockage des fichiers et d'accélérer le transfert de données. Lorsque vous lisez un fichier compressé, le **Studio Talend** doit le décompresser avant de pouvoir en alimenter le flux d'entrée. Notez que, lorsque le type de fichier à écrire est **Sequence File**, l'algorithme de compression est embarqué dans les fichiers du conteneur (les fichiers *part-*) de ce fichier de séquence. Ces fichiers peuvent être lus par un composant **Talend** comme le **tHDFSInput** dans des Jobs MapReduce et d'autres applications comprenant le format du fichier de séquence. Lorsque le type est **Text File**, les fichiers de sortie sont accessibles avec un utilitaire standard de compression prenant en charge les fichiers *bzip2* ou *gzip*. |
| Include header | Cochez cette case pour écrire en sortie l'en-tête des données. Ce champ n'est pas disponible pour un fichier Sequence. |

**Advanced settings**

| Hadoop properties | Le **Studio Talend** utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l'exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut. |
• Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste Property type, vous passiez de Repository à Built-in.

Pour plus d'informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d'Apache Hadoop sur http://hadoop.apache.org/docs en sélectionnant la version de la documentation souhaitée. À titre d'exemple, les liens vers certaines propriétés sont listés ci-après :

• Généralement, les propriétés relatives à HDFS peuvent être trouvées dans le fichier hdfs-default.xml correspondant à votre distribution, comme par exemple http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais).

• Apache fournit également une page listant toutes les propriétés relatives à Hive : http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml (en anglais).

**Variables globales**

Variables globales

**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant requiert un lien d’entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job.</td>
</tr>
</tbody>
</table>
Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le **Studio Talend**. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le **Studio Talend** et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est *lib\MapRClient.dll* dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

  Si vous n’ajoutez pas de librairie, il est possible que vous rencontrez l’erreur suivante : *no MapRClient in java.library.path*.

- Configurez l’argument **-Djava.library.path**, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]** dans le menu **Window**. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.
Limitation

La version 1.6+ de JRE est requise pour exécuter le composant.

Scénario associé

- Pour un scénario associé, consultez : Scénario 1 : Ecrire des données dans un fichier délimité à la page 1172,
- ou Scénario : Traiter des données avec le système de fichiers distribués Hadoop à la page 1594.
### tHDFSOutputRaw

Ce composant transmet des données de différents formats, comme des données hiérarchiques, dans une colonne, dans un système de fichiers HDFS donné.

Le tHDFSOutputRaw reçoit une colonne du flux d’entrée et écrit les données dans HDFS.

### Propriétés du tHDFSOutputRaw Standard

Ces propriétés sont utilisées pour configurer le tHDFSOutputRaw s'exécutant dans le framework de Jobs Standard.

Le composant tHDFSOutputRaw Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

#### Basic settings

| Property type | Peut être Built-In ou Repository.  
|               | **Built-In** : propriétés utilisées ponctuellement.  
|               | **Repository** : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.  

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
|                       | Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur **Edit schema** afin de visualiser le schéma.  
|                       | • **View schema** : sélectionnez cette option afin de voir le schéma.  
|                       | • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
|                       | • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].  

| Use an existing connection | Cochez cette case et, dans la liste **Component List**, cliquez sur le composant HDFS de connexion duquel vous souhaitez réutiliser les informations de connexion précédemment définies.  

Notez que lorsqu'un Job contient un Job parent et un Job enfant, la liste **Component List** présente uniquement les composants de connexion du Job du même niveau.

**Distribution**

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option **Custom** vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les Jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend**. **Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une
telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
</table>
| Use kerberos authentication | Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le “principal name” de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.  
  • Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans **Connexion sécurisée à MapR** à la page 1745.  
  Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.  
  Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez. |
| Use a keytab to authenticate | Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**.  
  Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.  
  Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est **user1** et le principal à utiliser est **guest**. Dans cette situation, |
<table>
<thead>
<tr>
<th><strong>assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NameNode URI</strong></td>
</tr>
<tr>
<td><strong>Use Datanode hostname</strong></td>
</tr>
<tr>
<td><strong>User name</strong></td>
</tr>
<tr>
<td><strong>Group</strong></td>
</tr>
<tr>
<td><strong>File Name</strong></td>
</tr>
</tbody>
</table>
| **Action** | Sélectionnez une opération dans HDFS :  
**Create** : crée un fichier du nom défini dans le champ **File Name**.  
**Overwrite** : écrase les données dans le fichier spécifié dans le champ **File Name**.  
**Append** : insère les données dans le fichier spécifié dans le champ **File Name**. Le fichier spécifié est automatiquement créé s’il n’existe pas. |
| **Custom encoding** | Il est possible de rencontrer des problèmes d’encodage lorsque vous traitez les données stockées. Dans ce cas, cochez cette case pour afficher la liste **Encoding**.  
Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.  
Cette option est indisponible pour les fichiers de type *Sequence*. |
| **Compression** | Cochez la case **Compress the data** afin de compresser les données de sortie. |
Hadoop fournit différents formats de compression permettant de réduire l’espace nécessaire au stockage des fichiers et d’accélérer le transfert de données. Lorsque vous lisez un fichier compressé, le Studio Talend doit le décompresser avant de pouvoir en alimenter le flux d’entrée.

Notez que, lorsque le type de fichier à écrire est **Sequence File**, l’algorithme de compression est embarqué dans les fichiers du conteneur (les fichiers *part-*) de ce fichier de séquence. Ces fichiers peuvent être lus par un composant Talend comme le **tHDFSInput** dans des Jobs MapReduce et d’autres applications comprenant le format du fichier de séquence. Lorsque le type est **Text File**, les fichiers de sortie sont accessibles avec un utilitaire standard de compression prenant en charge les fichiers *bzip2* ou *gzip*.

### Die on error

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Reject**.

### Advanced settings

#### Hadoop properties

Le Studio Talend utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :

tHDFSOutputRaw


**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Variables globales

| Variables globales | **FILENAME_PATH** : chemin du fichier d’entrée. Cette variable est une variable **After** et retourne une chaîne de caractères.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |
| Variables globales | **FILENAME_PATH** : chemin du fichier d’entrée. Cette variable est une variable **After** et retourne une chaîne de caractères.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant nécessite un composant d’entrée fournissant les données d’une colonne. Cette colonne doit se nommer **content** et doit être de type **Object**.  
Par exemple, vous pouvez :  
- utiliser un **tConvertType** pour convertir une colonne de **String** à **Object**,  
- ou utiliser un **tJavaRow** pour ajouter les données à traiter à l’objet **globalMap**, afin que les données soient disponibles comme variable globale pour les autres composants, comme le **tFixedFlowInput** pour construire cette colonne requise.  
Pour plus d’informations concernant le composant **tHMap**, consultez **tHMap**.  
Pour plus d’informations concernant le composant **tConvertType**, consultez **tConvertType** à la page 526.  
Pour plus d’informations concernant le composant **tJavaRow**, consultez **tJavaRow** à la page 1964.  
Pour plus d’informations concernant le composant **tFixedFlowInput**, consultez **tFixedFlowInput** à la page 1267. |
| Règle d’utilisation | Ce composant nécessite un composant d’entrée fournissant les données d’une colonne. Cette colonne doit se nommer **content** et doit être de type **Object**.  
Par exemple, vous pouvez :  
- utiliser un **tConvertType** pour convertir une colonne de **String** à **Object**,  
- ou utiliser un **tJavaRow** pour ajouter les données à traiter à l’objet **globalMap**, afin que les données soient disponibles comme variable globale pour les autres composants, comme le **tFixedFlowInput** pour construire cette colonne requise.  
Pour plus d’informations concernant le composant **tHMap**, consultez **tHMap**.  
Pour plus d’informations concernant le composant **tConvertType**, consultez **tConvertType** à la page 526.  
Pour plus d’informations concernant le composant **tJavaRow**, consultez **tJavaRow** à la page 1964.  
Pour plus d’informations concernant le composant **tFixedFlowInput**, consultez **tFixedFlowInput** à la page 1267. |
Pour plus d’informations concernant l’utilisation des variables globales, consultez la section décrivant l’utilisation des contextes et variables dans le Guide utilisateur du Studio Talend.

**Dynamic settings**

Cliquez sur le bouton `[+]` pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prédéfinies dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Prérequis**

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend.

La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées `MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native`. Par exemple, pour Windows, la bibliothèque est `lib\MapRClient.dll` dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : `no MapRClient in java.library.path`.

- Configurez l’argument `-Djava.library.path`, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque.
native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Scénario associé

Une fois la connexion à HDFS correctement configurée pour ce composant, il fonctionne exactement comme le tFileOutputRaw.

Pour plus d’informations concernant le tFileOutputRaw, consultez tFileOutputRaw à la page 1214.
tHDFSProperties

Ce composant permet de créer un flux d’une ligne affichant les propriétés d’un fichier traité dans HDFS.

**Propriétés du tHDFSProperties Standard**

Ces propriétés sont utilisées pour configurer le tHDFSProperties s’exécutant dans le framework de Jobs Standard.

Le composant tHDFSProperties Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Property type | Peut être Built-in ou Repository.  
|               | *Built-in* : Propriétés utilisées ponctuellement.  
|               | *Repository* : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.  
| Use an existing connection | Cochez cette case et, dans la liste Component List, cliquez sur le composant HDFS de connexion duquel vous souhaitez réutiliser les informations de connexion précédemment définies.  
|               | Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.  
| Distribution | Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
|               | • Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).  
|               | • L’option Custom vous permet de vous connecter à un cluster différente des clusters de la liste,
par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les Jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

| Hadoop version | Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez. |
| Use kerberos authentication | Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le “principal name” de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre |
identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

| Use a keytab to authenticate | Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser. |

| NameNode URI | Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée *masternode* comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé. |

| User name | Le champ **User name** est disponible lorsque vous n’utilisez pas Kerberos pour vous authentifier. Dans ce champ, saisissez votre identifiant pour cette distribution. Si vous laissez le champ vide, le nom de la machine hébergeant le Studio sera utilisé. |

| Group | Identifiant de l’utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ peut être disponible ou indisponible selon la distribution que vous utilisez. |
### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur **Edit schema** afin de visualiser le schéma.

### File

Parcourez votre système ou saisissez le chemin d'accès aux données à utiliser dans le système de fichiers.

### Get file checksum

Cochez cette case pour générer et écrire en sortie les informations MD5 du fichier traité.

Notez que cette somme de contrôle est pour HDFS uniquement et n’est pas un véritable hachage MD5 pouvant être comparé à la valeur MD5 obtenue avec un composant comme le `tFileInputProperties`. Pour plus d’informations concernant ce composant, consultez `tFileInputProperties` à la page 1134.

### Advanced settings

#### Hadoop properties

Le Studio Talend utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :


- Apache fournit également une page listant toutes les propriétés relatives à Hive : [http://hadoop.apache](http://hadoop.apache)
## Variables globales

| Variables globales | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

## Utilisation

| Règle d’utilisation | Le **tHDFSProperties** peut être utilisé en standalone ou envoyer les informations générées au composant suivant. |
| Dynamic settings | Cliquez sur le bouton [*] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez. Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les |
| Prérequis | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.  
• Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\ hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.  
• Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR. Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez. |

| Limitation | La version 1.6+ de JRE est requise pour exécuter le composant. |

**Scénario associé**

Pour un scénario associé, consultez :

• **Procédure** à la page 1221.

• **Scénario : Effectuer une boucle sur un répertoire HDFS** à la page 1616.
**tHDFSPut**

Ce composant se connecte au système de fichiers distribués Hadoop pour charger de gros fichiers avec une performance optimisée.

Le tHDFSPut copie des fichiers d’un répertoire défini par l’utilisateur et les colle dans un système de fichiers distribué Hadoop donné (HDFS). Si nécessaire, il renomme ces fichiers.

**Propriétés du tHDFSPut Standard**

Ces propriétés sont utilisées pour configurer le tHDFSPut s’exécutant dans le framework de Jobs Standard.

Le composant tHDFSPut Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th></th>
</tr>
</thead>
</table>
| **Property type** | Peut être **Built-in** ou **Repository**.  
**Built-in** : Propriétés utilisées ponctuellement.  
**Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th></th>
</tr>
</thead>
</table>
| **Use an existing connection** | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste **Component List** présente uniquement les composants de connexion du Job du même niveau. |

<table>
<thead>
<tr>
<th>Distribution</th>
<th></th>
</tr>
</thead>
</table>
| **Distribution** | Sélectionnez dans la liste le cluster que vous utiliserez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
• Si disponible dans la liste de **Distribution**, l’option **Microsoft HD Insight** vous permet d’utiliser un cluster Microsoft HD Insight. Dans cette optique, vous devez configurer les connexions au cluster HD Insight et au service Windows Azure Storage du cluster dans les zones affichées. Pour plus d’informations concernant ces paramètres, recherchez Configurer manuellement la connexion, sur Talend Help Center ([https://help.talend.com](https://help.talend.com)).  
• Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center ([https://help.talend.com](https://help.talend.com)). |
- L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use kerberos authentication</td>
<td>Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le &quot;principal name&quot; de Kerberos pour le NameNode</td>
</tr>
</tbody>
</table>
Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

| Use a keytab to authenticate | Cochez la case Use a keytab to authenticate pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser. |

| NameNode URI | Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée masternode comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé. |

| User name | Le champ User name est disponible lorsque vous n’utilisez pas Kerberos pour vous authentifier. Dans ce champ, saisissez votre identifiant pour cette distribution. Si vous laissez le champ vide, le nom de la machine hébergeant le Studio sera utilisé. |

| Group | Identifiant de l’utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ |
peut être disponible ou indisponible selon la distribution que vous utilisez.

<table>
<thead>
<tr>
<th>Local directory</th>
<th>Dossier local où sont stockés les fichiers obtenus de HDFS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDFS directory</td>
<td>Parcourez votre système ou saisissez le chemin d'accès aux données à utiliser dans le système de fichiers.</td>
</tr>
<tr>
<td>Overwrite file</td>
<td>Option permettant d'écraser ou non les fichiers avec le nouveau fichier.</td>
</tr>
</tbody>
</table>

**Use Perl5 Regex Expression as Filemask**

Cochez cette case si vous souhaitez utiliser les expressions régulières Perl5 comme filtres de fichiers dans le champ **Files**. Cela est utile lorsque le nom du fichier à utiliser contient des caractères spéciaux comme des parenthèses.

Pour plus d’informations concernant la syntaxe des expressions régulières Perl5, consultez [Perl5 Regular Expression Syntax](en anglais).

**Files**

Dans la zone **Files**, les champs à remplir sont :
- **File mask** : saisissez le nom du fichier qui sera sélectionné dans HDFS. Vous pouvez utiliser des expressions régulières.
- **New name** : renommez le fichier obtenu.

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Hadoop properties**

Le *Studio Talend* utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- Notez que, si vous utilisez les métadonnées stockées centralement dans le *Repository*, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste *Property type*, vous passiez de *Repository* à *Built-in*.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la
documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d'exemple, les liens vers certaines propriétés sont listés ci-après :


### Variables globales

- **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable *After* et retourne un entier.
- **TRANSFER_MESSAGES** : informations transférées du fichier. Cette variable est une variable *After* et retourne une chaîne de caractères.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

Le composant combine la connexion HDFS et l’extraction de données, de cette façon il est souvent utilisé comme simple composant pour déplacer des données d’un répertoire local donné à HDFS.

A la différence des composants *tHDFSInput* et *tHDFSOutput*, il s’exécute en standalonde et ne génère pas de flux d’entrée ou de sortie pour les autres composants.

Ce composant est généralement connecté à un Job par un lien *OnSubjobOk* ou *OnComponentOk*, selon le contexte.
Cliquez sur le bouton [+ pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.


Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Prérequis
Pour plus d'informations concernant l'installation d'une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

| Limitation | La version 1.6+ de JRE est requise pour exécuter le composant. |

**Scénario associé**

Pour un scénario associé, consultez Scénario : Traiter des données avec le système de fichiers distribués Hadoop à la page 1594.
Ce composant renomme le(s) fichier(s) ou répertoire(s) sélectionné(s) dans HDFS.

### Propriétés du tHDFSRename Standard

Ces propriétés sont utilisées pour configurer le tHDFSRename s'exécutant dans le framework de Jobs Standard.

Le composant tHDFSRename Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

#### Basic settings

| Property type       | Peut être **Built-in** ou **Repository**.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
</tbody>
</table>

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste <strong>Component List</strong> présente uniquement les composants de connexion du Job du même niveau.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez. Parmi ces options, les suivantes nécessitent une configuration spécifique.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Si vous sélectionnez <strong>Amazon EMR</strong>, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
<tr>
<td></td>
<td>• L’option <strong>Custom</strong> vous permet de vous connecter à un cluster différente des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.</td>
</tr>
</tbody>
</table>
1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend**. **Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utiliserez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use kerberos authentication</td>
<td>Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le &quot;principal name&quot; de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.</td>
</tr>
</tbody>
</table>
- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l'authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l'esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d'utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d'Hadoop à laquelle vous vous connectez.

### Use a keytab to authenticate

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l'aide d'un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est **user1** et le principal à utiliser est **guest**. Dans cette situation, assurez-vous que **user1** a les droits de lecture pour le fichier keytab à utiliser.

### NameNode URI

Saisissez l'URI du NameNode Hadoop, nœud maître d'un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée **masternode** comme NameNode, son emplacement est `hdfs://masternode:portnumber`. Si vous utilisez WebHDFS, l'emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL, le schéma d'URI doit être `swebhdfs` et vous devez utiliser un `tLibraryLoad` dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

### Username

Le champ **User name** est disponible lorsque vous n'utilisez pas Kerberos pour vous authentifier. Dans ce champ, saisissez votre identifiant pour cette distribution. Si vous laissez le champ vide, le nom de la machine hébergeant le Studio sera utilisé.

### Group

Identifiant de l'utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ peut être disponible ou indisponible selon la distribution que vous utilisez.
<table>
<thead>
<tr>
<th><strong>HDFS directory</strong></th>
<th>Parcourez votre système ou saisissez le chemin d'accès aux données à utiliser dans le système de fichiers.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overwrite file</strong></td>
<td>Sélectionnez l'option souhaitée pour écraser ou non le fichier existant et le remplacer par le nouveau.</td>
</tr>
</tbody>
</table>
| **Files** | Cliquez sur le bouton [+] pour ajouter des lignes à utiliser comme filtres :  
* **Filemask** : saisissez le nom du fichier ou du masque de fichier à l'aide de caractères de remplacement ou d'expressions régulières.  
* **New name** : saisissez le nom à donner au fichier HDFS après le transfert. |
| **Die on error** | Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. |

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>tStatCatcher Statistics</strong></th>
<th>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau du composant.</th>
</tr>
</thead>
</table>
| **Hadoop properties**     | Le *Studio Talend* utilise une configuration par défaut pour son moteur, afin d'effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l'exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.  
* Notez que, si vous utilisez les métadonnées stockées centralement dans le repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste Property type, vous passiez de repository à Built-in.  
Pour plus d'informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d'Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d'exemple, les liens vers certaines propriétés sont listés ci-après :  
* Apache fournit également une page listant toutes les propriétés relatives à Hive : [http://hadoop.apache](http://hadoop.apache) |
### Variables globales

<table>
<thead>
<tr>
<th>Variables globales</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_FILE</strong></td>
<td>Indique le nombre de fichiers traités. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>CURRENT_STATUS</strong></td>
<td>résultats d'exécution du composant. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option. Une variable <em>Flow</em> fonctionne durant l'exécution d'un composant. Une variable <em>After</em> fonctionne après l'exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Ce composant peut être utilisé en standalone.</td>
</tr>
</tbody>
</table>

Cliquez sur le bouton `[+]` pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant. La table *Dynamic settings* est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue *Basic settings*. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue *Basic settings* devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les
La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

La version 1.6+ de JRE est requise pour exécuter le composant.

Scénario associé

Pour un scénario associé, consultez Scénario : Traiter des données avec le système de fichiers distribués Hadoop à la page 1594.
**tHDFSRowCount**

Ce composant lit un fichier dans HDFS ligne par ligne afin de déterminer le nombre de lignes qu’il contient.

Le tHDFSRowCount compte le nombre de lignes d’un fichier donné dans HDFS. Si le fichier à traiter est un fichier Hadoop de type séquence ou un ensemble de données volumineux, il est recommandé d’utiliser un composant tAggregateRow pour compter les enregistrements.

**Propriétés du tHDFSRowCount Standard**

Ces propriétés sont utilisées pour configurer le tHDFSRowCount s’exécutant dans le framework de Jobs Standard.

Le composant tHDFSRowCount Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

|---------------|-------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|

**Use an existing connection**

Cochez cette case et, dans la liste Component List, cliquez sur le composant HDFS de connexion duquel vous souhaitez réutiliser les informations de connexion précédemment définies.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.

**Distribution**

 Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez. Parmi ces options, les suivantes nécessitent une configuration spécifique.

- Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).
• L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use kerberos authentication</td>
<td>Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le &quot;principal name&quot; de Kerberos pour le NameNode</td>
</tr>
</tbody>
</table>
Dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases *Force MapR ticket authentication* et *Use Kerberos authentication*. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

| **Use a keytab to authenticate** | Cochez la case *Use a keytab to authenticate* pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ *Principal* et le chemin d’accès au fichier keytab dans le champ *Keytab*. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser. |
| **NameNode URI** | Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée *masternode* comme NameNode, son emplacement est *hdfs://masternode:portnumber*. Si vous utilisez WebHDFS, l’emplacement doit être *webhdfs://masternode:portnumber*. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être *swebhdfs* et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé. |
| **User name** | Le champ *User name* est disponible lorsque vous n’utilisez pas Kerberos pour vous authentifier. Dans ce champ, saisissez votre identifiant pour cette distribution. Si vous laissez le champ vide, le nom de la machine hébergeant le Studio sera utilisé. |
| **Group** | Identifiant de l’utilisateur et nom du groupe sous lesquels les instances HDFS ont été lancées. Ce champ |
File name | Parcourrez votre système ou saisissez le chemin d'accès aux données à utiliser dans le système de fichiers.
---|---
Row separator | Saisissez le séparateur utilisé pour identifier la fin d'une ligne.
Ignore empty rows | Cochez cette case pour ignorer les lignes vides.
Encoding | Sélectionnez l'encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.
Compression | Cochez la case Uncompress the data pour décompresser les données d'entrée.

Hadoop fournit différents formats de compression permettant de réduire l'espace nécessaire au stockage des fichiers et d'accélérer le transfert de données. Lorsque vous lisez un fichier compressé, le Studio Talend doit le décompresser avant de pouvoir en alimenter le flux d'entrée.

**Advanced settings**

**Hadoop properties** | Le Studio Talend utilise une configuration par défaut pour son moteur, afin d'effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.
---|---

- Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste Property type, vous passiez de Repository à Built-in.

Pour plus d'informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d'Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d'exemple, les liens vers certaines propriétés sont listés ci-après :

**tHDFSRowCount**


### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau du composant.

<table>
<thead>
<tr>
<th>Variables globales</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variables globales</strong></td>
</tr>
<tr>
<td><strong>COUNT</strong> : nombre de lignes dans un fichier. Cette variable est une variable <strong>Flow</strong> et retourne un nombre entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**

Le **tHDFSRowCount** est un composant standalone. Il peut être utilisé avec un lien **OnSubjobOk** vers un **tJava** afin de retourner le nombre de lignes.

Le code valide pour que le **tJava** obtienne ce compte est, par exemple, le suivant :

```
System.out.print(((Integer)globalMap.get("tHDFSRowCount_1_COUNT")));
```

Dans cet exemple, **tHDFSRowCount_1** est le nom d’un des composants d’un Job. Cela peut varier à travers les différents scénarios. **COUNT** est la variable globale du **tHDFSRowCount** et représente le flux de type Integer du compte de lignes.

Pour plus d’informations concernant le nommage d’un composant ou l’utilisation des variables globales dans un Job, consultez le Guide utilisateur du Studio Talend.

**Dynamic settings**

Cliquez sur le bouton `[+]` pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre
connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un Studio Talend indépendant.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontrez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.
**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tHiveClose

Ce composant ferme une connexion à la base de données Hive.
Le tHiveClose ferme la connexion à une base de données connectée.

Propriétés du tHiveClose Standard

Ces propriétés sont utilisées pour configurer le tHiveClose s’exécutant dans le framework de Jobs Standard.
Le composant tHiveClose Standard appartient aux familles Big Data et Databases.
Le composant de ce framework est toujours disponible.

Basic settings

| Component list | S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tHiveConnection dans la liste. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Hive, notamment le tHiveConnection, car le tHiveConnection vous permet d’ouvrir une connexion pour la transaction à effectuer. |
### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.


### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le **Studio Talend**. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le **Studio Talend** et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les bibliothèques du client MapR correspondant à chaque OS peuvent être trouvées **MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native**. Par exemple, pour Windows, la bibliothèque est **lib\MapRClient.dll** dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

  Si vous n’ajoutez pas de bibliothèque, il est possible que vous rencontriez l’erreur suivante : 

  ```
  no MapRClient in java.library.path.
  ```

- Configurez l’argument **-Djava.library.path**, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]** dans le menu **Window**. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des
Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tHiveConnection

Ce composant établit une connexion Hive à réutiliser dans d’autres composants Hive de votre Job.
Le tHiveConnection ouvre une connexion vers une base de données Hive.

Propriétés du tHiveConnection Standard

Ces propriétés sont utilisées pour configurer le tHiveConnection s’exécutant dans le framework de Jobs Standard.
Le composant tHiveConnection Standard appartient aux familles Big Data, Databases et ELT.
Le composant de ce framework est toujours disponible.

Basic settings
Configuration de la connexion :
• Lorsque vous utilisez ce composant avec Google Dataproc :

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster identifier</td>
<td>Saisissez l’ID de votre cluster Dataproc à utiliser.</td>
</tr>
<tr>
<td>Region</td>
<td>Saisissez les régions dans lesquelles sont utilisées les ressources de calcul et dans lesquelles sont stockées et traitées les données. Si vous n'avez pas besoin de spécifier une région en particulier, laissez la valeur par défaut global. Pour plus d’informations relatives aux régions disponibles et aux zones de chaque groupe de région, consultez Regions and Zones (en anglais).</td>
</tr>
<tr>
<td>Google Storage staging bucket</td>
<td>Comme un Job Talend nécessite ses fichiers .jar dépendants pour être exécuté, spécifiez le répertoire Google Storage dans lequel ces fichiers .jar sont transférés afin que votre Job accède à ces fichiers lors de l’exécution. Le répertoire à saisir doit se terminer par une barre oblique (/). Si le répertoire n’existe pas, un répertoire est créé à la volée mais le bucket à utiliser doit déjà exister.</td>
</tr>
<tr>
<td>Database</td>
<td>Renseignez ce champ avec le nom de la base de données.</td>
</tr>
<tr>
<td>Provide Google Credentials in file</td>
<td>Lorsque vous lancez votre Job à partir d’une machine donnée sur laquelle Google Cloud SDK a été installé et vous a autorisé à utiliser vos identifiants de compte utilisateur pour accéder à Google Cloud Platform,</td>
</tr>
</tbody>
</table>
ne cochez pas cette case. Dans cette situation, cette machine est souvent votre machine locale.


- Lorsque vous utilisez ce composant avec HDInsight :

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HDInsight configuration</td>
<td>Saisissez les informations d’authentification du cluster HD Insight à utiliser.</td>
</tr>
<tr>
<td>Database</td>
<td>Renseignez ce champ avec le nom de la base de données.</td>
</tr>
</tbody>
</table>

- Lorsque vous utilisez les autres distributions :

<table>
<thead>
<tr>
<th>Connection mode</th>
<th>Sélectionnez un mode de connexion dans la liste. Les options varient en fonction de la distribution que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hive server</td>
<td>Sélectionnez le serveur Hive sur lequel vous souhaitez que le Job utilisant ce composant exécute des requêtes dans Hive. La liste Hive server est disponible uniquement lorsque la distribution Hadoop à utiliser, par exemple HortonWorks Data Platform V1.2.0 (Bimota) supporte HiveServer2. Vous pouvez sélectionner HiveServer2 (Hive 2), le serveur supportant mieux les connexions simultanées de différents clients que HiveServer (Hive 1).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Renseignez ce champ avec le nom de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Remarque : Ce champ n’est pas disponible lorsque vous sélectionnez l’option Embedded dans la liste Connection mode.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Données d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Use kerberos authentication</strong></td>
<td>Si vous accédez au Metastore de Hive avec une sécurité Kerberos, cochez cette case et saisissez ensuite les paramètres appropriés dans les champs qui s’affichent.</td>
</tr>
<tr>
<td></td>
<td>• Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.</td>
</tr>
<tr>
<td></td>
<td>Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.</td>
</tr>
<tr>
<td></td>
<td>Les valeurs des paramètres suivants peuvent être trouvées dans le fichier hive-site.xml du système Hive utilisé.</td>
</tr>
<tr>
<td></td>
<td>1. <strong>Hive principal</strong> utilise la valeur de hive.metastore.kerberos.principal. C’est le principal du service du Metastore de Hive.</td>
</tr>
<tr>
<td></td>
<td>2. <strong>HiveServer2 local user principal</strong> utilise la valeur de hive.server2.authentication.kerberos.principal.</td>
</tr>
<tr>
<td></td>
<td>3. <strong>HiveServer2 local user keytab</strong> utilise la valeur de hive.server2.authentication.kerberos.keytab</td>
</tr>
</tbody>
</table>
4. **Metastore URL** utilise la valeur de `javax.jdo.option.ConnectionURL`. C'est la chaîne JDBC de connexion au Metastore de Hive.

5. **Driver class** utilise la valeur de `javax.jdo.option.ConnectionDriverName`. C'est le nom du pilote de la connexion JDBC.

6. **Username** utilise la valeur de `javax.jdo.option.ConnectionUserName`. Ce paramètre, ainsi que le paramètre **Password**, sont utilisés pour les informations de connexion de l'utilisateur au Metastore de Hive.

7. **Password** utilise la valeur de `javax.jdo.option.ConnectionPassword`.

Cette case est disponible ou indisponible selon la distribution d'Hadoop à laquelle vous vous connectez.

---

### Use a keytab to authenticate

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l'aide d'un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est `user1` et le principal à utiliser est `guest`. Dans cette situation, assurez-vous que `user1` a les droits de lecture pour le fichier keytab à utiliser.

---

### Use SSL encryption

Cochez cette case pour activer la connexion chiffrée SSL ou TLS.

Les champs qui s'affichent ensuite fournissent les informations d'authentification :

- Dans le champ **Trust store path**, saisissez le chemin ou parcourrez votre système jusqu'au fichier TrustStore à utiliser. Par défaut, les types TrustStore supportés sont JKS et PKCS 12.

- Pour saisir le mot de passe, cliquez sur le bouton `...` à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Cette fonctionnalité n’est disponible que pour HiveServer2 en mode **Standalone** pour les distributions suivantes :

- Hortonworks Data Platform 2.0 +
- Cloudera CDH4 +
- Pivotal HD 2.0 +

Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d’un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du de l’exécution) :

1. Cochez la case **Set resourcemanager scheduler address** et saisissez l’adresse de l’ordonnanceur (Scheduler) dans le champ qui apparaît.

2. Cochez la case **Set jobhistory address** et saisissez l’emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory.

3. Cochez la case **Set staging directory** et saisissez le chemin d’accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l’exécution de programmes. Ce répertoire se trouve sous la propriété `yarn.app.mapreduce.am.staging-dir` dans les fichiers de configuration, notamment les fichiers `yarn-site.xml` et `mapred-site.xml` de votre distribution.

4. Allouez des volumes de mémoire aux calculs **Map** et **Reduce** et au service **ApplicationMaster** de YARN en cochant la case **Set memory** dans la vue **Advanced settings**.

5. Cochez la case **Set Hadoop user** et saisissez le nom de l’utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d’écriture, ce champ vous permet d’exécuter le Job directement avec l’utilisateur ayant les droits d’accès appropriés au fichier ou répertoire à traiter.

6. Cochez la case **Use datanode hostname** pour permettre au Job d’accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété `dfs.client.use.datanode.hostname` à `true`. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.


| Set NameNode URI | Cochez cette case et, dans le champ qui s’affiche, saisissez l’URI du NameNode Hadoop, le nœud maître d’un système Hadoop. Par exemple, si vous avez choisi une machine nommée `masternode` en tant que NameNode, l’emplacement est `hdfs://masternode`.

---

- Amazon EMR 4.0.0 +
Si vous utilisez WebHDFS, l'emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL, le schéma d'URI doit être `swebhdfs` et vous devez utiliser un `tLibraryLoad` dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.


Les autres propriétés :

| Property type | Peut être **Built-In** ou **Repository**.  
**Built-In** : propriétés utilisées ponctuellement.  
**Repository** : sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Distribution**

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.

- Si disponible dans la liste de **Distribution**, l'option **Microsoft HD Insight** vous permet d'utiliser un cluster Microsoft HD Insight. Dans cette optique, vous devez configurer les connexions au cluster HD Insight et au service Windows Azure Storage du cluster dans les zones affichées. Pour plus d'informations concernant ces paramètres, recherchez Configurer manuellement la connexion, sur Talend Help Center ([https://help.talend.com](https://help.talend.com))..

- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d'informations concernant la configuration d'un cluster Amazon EMR dans Talend Help Center ([https://help.talend.com](https://help.talend.com)).

- L'option **Custom** vous permet de vous connecter à un cluster différente des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre
connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend**. **Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

### Hive version

Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.

### Inspect the classpath for configurations

Cochez cette case pour permettre au composant de vérifier les fichiers de configuration dans le répertoire configuré pour la variable `$HADOOP_CONF_DIR` et de lire directement les paramètres de ces fichiers dans le répertoire. Cette fonctionnalité vous permet de modifier facilement la configuration Hadoop afin que le composant puisse passer d’un environnement à un autre, comme par exemple pour passer d’un environnement test à un environnement production.

Dans ce cas, les champs ou les options utilisée pour configurer la connexion Hadoop et/ou la sécurité Kerberos sont masqués.

Si vous souhaitez utiliser certains paramètres comme les paramètres Kerberos mais que ces paramètres ne sont pas inclus dans les fichiers de configuration Hadoop, vous devez créer un fichier appelé `talend-site.xml` et mettre ce fichier dans le répertoire défini dans `$HADOOP_CONF_DIR`. Le fichier `talend-site.xml` doit se présenter comme suit :

```xml
<!-- Put site-specific property overrides in this file. -->
```
Les paramètres lus depuis ces fichiers de configuration écrasent ceux utilisés par défaut dans le Studio. Lorsqu’un paramètre n’existe pas dans ces fichiers de configuration, le paramètre par défaut est utilisé.

Notez que cette option est disponible uniquement dans le mode **Standalone** de Hive avec **Hive 2**.
| **Use or register a shared DB Connection** | Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un composant **tRunJob** ayant une de ces options activée fera échouer votre Job. |
| **Execution engine** | Cochez cette case et, dans la liste déroulante, sélectionnez le framework à utiliser pour exécuter le Job.

Cette liste est disponible lorsque vous utilisez le mode **Embedded** pour la connexion et distribution Hive avec laquelle vous travaillez, parmi les suivantes :
- Hortonworks : V2.1 et V2.2.
- MapR : V4.0.1.
- **Custom** : cette option vous permet de vous connecter à une distribution supportant mais non officiellement supportée par Talend.

Avant d’utiliser Tez, vérifiez que votre cluster Hadoop supporte Tez. Vous devez configurer l’accès aux bibliothèques Tez correspondantes via la vue **Advanced settings** de ce composant.

Pour plus d’informations concernant Hive avec Tez, consultez la documentation Apache à l’adresse https://cwiki.apache.org/confluence/display/Hive/Hive+on+Tez (en anglais). Des exemples vous sont présentés afin d’expliquer comment Tez peut être utilisé pour optimiser les performances par rapport à MapReduce. |
| **Store by HBase** | Cochez cette case afin d’afficher les paramètres à configurer pour permettre aux composants Hive d’accéder aux tables HBase :
- Une fois l’accès configuré, vous pourrez utiliser, dans un **tHiveRow** et un **tHiveInput**, les instructions Hive QL permettant de lire et d’écrire des données dans HBase.
- Si vous utilisez l’authentification Kerberos, vous devez définir les principaux relatifs à HBase dans les champs correspondants qui sont affichés.

Pour plus d’informations à propos de cet accès concernant Hive et HBase, consultez la documentation de Apache Hive concernant l’intégration Hive/HBase. |
| **Zookeeper quorum** | Saisissez le nom ou l’URL du service Zookeeper utilisé pour coordonner les transactions entre votre Studio et votre base de données. Notez que, lorsque vous... |
configurez Zookeeper, vous pouvez avoir besoin de configurer explicitement la propriété `zookeeper.znode.parent` pour définir le chemin vers le nœud znode racine contenant tous les znodes créés et utilisés par votre base de données. Cochez la case `Set Zookeeper znode parent` afin de définir cette propriété.

### Zookeeper client port

Saisissez le numéro du port d'écoute client du service Zookeeper que vous utilisez.

### Define the jars to register for HBase

Cochez cette case pour afficher la table **Register jar for HBase**, dans laquelle vous pouvez enregistrer tout fichier Jar manquant, requis pour HBase, par exemple, Hive Storage Handler, enregistré par défaut avec votre installation Hive.

### Register jar for HBase

Cliquez sur le bouton `[+]` pour ajouter des lignes à la table, puis, dans la colonne `Jar name`, sélectionnez le(s) fichier(s) Jar à enregistrer. Dans la colonne `Jar path`, saisissez le chemin d'accès à ce(s) Jar(s).

### Advanced settings

#### Tez lib

Choisissez comment accéder aux bibliothèques de Tez :

- **Auto install** : lors de l'exécution, le Job charge et déploie les bibliothèques de Tez fournies par le Studio dans le répertoire spécifié dans le champ `Install folder in HDFS`, par exemple, `/tmp/usr/tez`.
  
  Si vous avez configuré la propriété `tez.lib.uris` dans la table des propriétés, ce répertoire écrase la valeur de la propriété lors de l'exécution. Les autres propriétés configurées dans la table sont toujours prises en compte.

- **Use exist** : le Job accède aux bibliothèques de Tez déjà déployées dans le cluster Hadoop à utiliser. Vous devez saisir le chemin d'accès pointant vers ces bibliothèques dans le champ `Lib path (folder or file)`.

- **Lib jar** : cette table s'affiche lorsque vous avez sélectionné **Auto install** dans la liste **Tez lib** et que vous utilisez une distribution personnalisée (**Custom**). Dans cette table, vous devez ajouter les bibliothèques de Tez à charger.

#### Hadoop properties

Le Studio Talend utilise une configuration par défaut pour son moteur, afin d'effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l'exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce
que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :


### Hive properties

Le **Studio Talend** utilise la configuration par défaut pour son moteur afin d’effectuer des opérations dans un base de données Hive. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez cette table avec la (les) propriété(s) à personnaliser. Ensuite, à l’exécution, la (les) propriété(s) personnalisée(s) écrasent celles par défaut. Pour plus d’informations concernant les propriétés dédiées à Hive, consultez [https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration](https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration) (en anglais).

- Si vous devez utiliser Tez pour exécuter votre Job Hive, ajoutez `hive.execution.engine` à la colonne **Properties** et `Tez` à la valeur **Value**, en entourant ces chaînes de caractères de guillemets doubles.
- Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

### Mapred job map memory mb et Mapred job reduce memory mb

Vous pouvez personnaliser les opérations map et reduce en cochant la case **Set memory**, pour configurer les allocations de mémoire pour ces opérations à effectuer par le système Hadoop.

Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations map et reduce dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux 1000, ce qui est normalement adapté pour l’exécution de ces opérations.
<table>
<thead>
<tr>
<th>Path separator in server</th>
<th>Laissez le champ <strong>Path separator in server</strong> tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d’autres termes, changez le séparateur si celui-ci n’est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case afin de collecter les données de log au niveau des composants.</td>
</tr>
</tbody>
</table>

### Global Variables

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé avec d’autres composants Hive, particulièrement avec **tHiveClose**.

Si le **Studio Talend** utilisé pour vous connecter à une base de données Hive fonctionne sous Windows, vous devez créer manuellement un dossier appelé *tmp* à la racine du disque où le **Studio Talend** est installé.

**Prérequis**

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le **Studio Talend**. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le **Studio Talend** et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées **MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native**. Par exemple, pour Windows, la bibliothèque est **lib\MapRClient.dll** dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).
Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante:
\[\text{no MapRClient in java.library.path.}\]

- Configurez l’argument `-Djava.library.path`, par exemple, dans la zone `Job Run VM arguments` de la vue `Run/Debug` de la boîte de dialogue `[Preferences]` dans le menu `Window`. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (`Data viewer`) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

### Connexion à une distribution Hadoop personnalisée

Comme expliqué dans le tableau des propriétés, lorsque vous sélectionnez l’option `Custom` dans la liste déroulante `Distribution`, vous pouvez vous connectez à une distribution qui n’est pas dans la liste `Distribution`.

Une fois l’option `Custom` sélectionnée, cliquez sur le bouton [Import custom definition] pour afficher la boîte de dialogue [Import custom definition], puis procédez comme suit :

**Procédure**

1. Selon votre cas, sélectionnez `Import from existing version` ou `Import from zip` afin de configurer la distribution Hadoop personnalisée à laquelle vous souhaitez vous connecter.


   - Sinon, sélectionnez `Import from existing version` pour importer une distribution de base officiellement supportée afin de la personnaliser à l’aide de l’assistant suivant.
Notez que les cases de cet assistant vous permettent de sélectionner le(s) élément(s) Hadoop que vous souhaitez importer. Selon le contexte dans lequel vous créez la connection, toutes les cases ne sont pas disponibles. Par exemple, si vous créez cette connexion pour un composant Hive, seule la case **Hive** est disponible.

2. Que vous ayez sélectionné **Import from existing version** ou **Import from zip**, vérifiez que chaque case à côté de l'élément Hadoop que vous souhaitez importer est cochée.

3. Cliquez sur **OK** et, dans la fenêtre d’avertissement, cliquez sur **Yes** pour accepter d’écraser toute configuration personnalisée des fichiers Jar précédemment implémentée.

La boîte de dialogue **[Custom Hadoop version definition]** devient active.
Cette boîte de dialogue liste les éléments Hadoop et les fichiers Jar que vous importez.

4. Si vous avez coché l’option **Import from zip**, cliquez sur **OK** afin de valider la configuration importée.

Si vous avez sélectionné l’option **Import from existing version**, afin d’importer une distribution de base, vous devez également importer des fichiers jar pour personnaliser cette distribution. Dans l’onglet de l’élément Hadoop que vous souhaitez personnaliser, par exemple l’onglet **HDFS/HCatalog/Oozie**, cliquez sur le bouton **[+]** pour ouvrir la boîte de dialogue **[Select libraries]**.

5. Sélectionnez l’option **External libraries** pour ouvrir sa vue.

6. Cliquez sur le bouton **Browse...** et parcourez votre système jusqu’au fichier Jar que vous souhaitez importer.

7. Cliquez sur **OK** pour valider les modifications et fermer la boîte de dialogue **[Select libraries]**.

Le fichier Jar sélectionné apparaît dans la liste de l’onglet de l’élément Hadoop configuré. Notez que si vous souhaitez partager la configuration personnalisée Hadoop avec un autre **Studio Talend**, vous pouvez exporter cette connexion personnalisée de la boîte de dialogue **[Custom Hadoop version definition]** en cliquant sur le bouton **.**

8. Dans la boîte de dialogue **[Custom Hadoop version definition]**, cliquez sur **OK** afin de valider la configuration personnalisée et retourner à la liste **Distribution** de la vue **Basic settings** du composant.
**Résultats**

Une fois la configuration de la distribution Hadoop personnalisée terminée vous pouvez, depuis la vue **Basic settings**, continuer de saisir d’autres paramètres requis pour la connexion.

Si la distribution Hadoop personnalisée à laquelle vous souhaitez vous connecter contient YARN, cochez la case **Use YARN** à côté de la liste **Distribution**.

Une vidéo est disponible à l’adresse suivante, pour expliquer, en prenant l’exemple de HDFS, comment configurer une connexion vers un cluster Hadoop personnalisé, également appelé distribution Hadoop non supportée : [How to add an unsupported Hadoop distribution to the Studio](en anglais).

**Scénario : Créer une table Hive partitionnée**

Ce scénario explique comment utiliser un composant **tHiveConnection**, un composant **tHiveCreateTable** et un **tHiveLoad** afin de créer une table Hive partitionnée et d’y écrire des données.

Notez que les composants **tHiveCreateTable** et **tHiveLoad** ne sont disponibles que si vous utilisez une des solutions **Talend** avec Big Data.

Les données d’exemple à utiliser dans ce scénario sont des informations relatives aux employés d’une entreprise et se présentent comme suit :

1;Lyndon;Fillmore;21-05-2008;US
2;Ronald;McKinley;15-08-2008
3;Ulysses;Roosevelt;05-10-2008
4;Harry;Harrison;23-11-2007
5;Lyndon;Garfield;19-07-2007
6;James;Quincy;15-07-2008
7;Chester;Jackson;26-02-2008
8;Dwight;McKinley;16-07-2008
9;Jimmy;Johnson;23-12-2007
10;Herbert;Fillmore;03-04-2008
Les informations contiennent le nom de certains employés et la date de leur enregistrement dans le système des ressources humaines. Comme ces employés travaillent pour la filiale américaine de l’entreprise, créez une partition US pour ces données.

Avant de commencer à reproduire ce scénario, assurez-vous d’avoir les droits d’accès appropriés à la base de données Hive à utiliser.

Notez que si vous utilisez le système d’exploitation Windows, vous devez créer un dossier tmp à la racine du disque où le studio est installé.

Procédez comme suit :

**Relier les composants**

**Procédure**

1. Dans la perspective **Integration** du studio, créez un Job vide depuis le nœud **Job Designs** du **Repository**.
   
   Pour plus d’informations concernant la création de Jobs, consultez le chapitre relatif dans le **Guide utilisateur du Studio Talend**.

2. Déposez un **tHiveConnection**, un **tHiveCreateTable** et un **tHiveLoad** dans l’espace de modélisation graphique.

3. Reliez-les à l’aide de liens **Trigger > OnSubjobOk**.

**Configurer la connexion à Hive**

**Pourquoi et quand exécuter cette tâche**

Configurer le tHiveConnection

**Procédure**

1. Double-cliquez sur le **tHiveConnection** pour ouvrir sa vue **Component**.

Pour plus d’informations concernant la création d’une connexion Hadoop dans le **Repository**, consultez le chapitre décrivant le nœud **Hadoop cluster** dans le **Guide de prise en main de Talend Open Studio for Big Data**.

3. Dans la zone **Version**, sélectionnez la distribution de Hadoop à utiliser, ainsi que sa version. Si votre distribution n’est pas dans la liste, sélectionnez **Custom** pour vous connecter à une distribution de Hadoop non officiellement supportée dans le studio.

Pour comprendre comment utiliser l’option **Custom** étape par étape, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

4. Dans la zone **Connection**, saisissez les paramètres de connexion à la base de données Hive à utiliser.


Configurer le tHiveConnection

Procédure

1. Double-cliquez sur le tHiveConnection pour ouvrir sa vue Component.

2. Dans la liste Property type, sélectionnez Built-in. Si vous avez créé la connexion à utiliser dans le Repository, sélectionnez Repository, cliquez sur le bouton pour ouvrir la boîte de dialogue [Repository content] et sélectionnez cette connexion. Ainsi, le studio réutilise les paramètres de connexion dans le Job.

Pour plus d'informations concernant la création d'une connexion Hadoop dans le Repository, consultez le chapitre décrivant le nœud Hadoop cluster dans le Guide de prise en main de Talend Open Studio for Big Data.

3. Dans la zone Version, sélectionnez la distribution de Hadoop à utiliser, ainsi que sa version. Si votre distribution n'est pas dans la liste, sélectionnez Custom pour vous connecter à une distribution de Hadoop non officiellement supportée dans le studio.

Pour comprendre comment utiliser l'option Custom étape par étape, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

4. Dans la zone Connection, saisissez les paramètres de connexion à la base de données Hive à utiliser.

5. Dans le champ Name node, saisissez l'emplacement du nœud maître, le NameNode, de la distribution à utiliser. Par exemple, saisissez talend-hdp-all:50300. Si vous utilisez WebHDFS, l'emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d'URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.


**Créer la table Hive**

**Définir le schéma**

**Procédure**

1. Double-cliquez sur le **tHiveCreateTable** pour ouvrir sa vue **Component**.

2. Cochez la case **Use an existing connection** et, dans la liste **Component list**, sélectionnez la connexion configurée dans le composant **tHiveConnection** utilisé dans ce Job.

3. Cliquez sur le bouton à côté du champ **Edit schema** pour ouvrir l’éditeur de schéma.

4. Cliquez quatre fois sur le bouton pour ajouter quatre lignes dans la colonne **Column**. Renommez-les respectivement *Id*, *FirstName*, *LastName* et *Reg_date*. 
Notez que vous ne pouvez pas utiliser les mots-clés réservés à Hive pour nommer les colonnes, comme par exemple *location* ou *date*.

5. Dans la colonne *Type*, sélectionnez le type de données dans chaque colonne. Dans ce scénario, *Id* est de type *Integer*, *Reg_date* est de type *Date* et les autres lignes sont de type *String*.

6. Dans la colonne *DB type*, sélectionnez le type Hive de chaque colonne correspondant au type de données défini. Par exemple, *Id* est de type *INT* et *Reg_date* est de type *TIMESTAMP*.

7. Dans la colonne *Data pattern*, définissez le modèle correspondant à celui des données brutes. Dans cet exemple, utilisez le modèle par défaut.

8. Cliquez sur **OK** pour valider ces modifications.

**Configurer les paramètres de la table**

**Procédure**

1. Dans le champ *Table name*, saisissez le nom de la table Hive à créer. Dans ce scénario, saisissez *employees*.

2. Dans la liste *Action on table*, sélectionnez *Create table if not exists*.

3. Dans la liste *Format*, sélectionnez le format des données pour lequel la table Hive est créée. Dans ce scénario, sélectionnez *TEXTFILE*.

4. Cochez la case *Set partitions* pour ajouter la partition *US* comme expliqué au début de ce scénario.

   Pour définir cette partition, cliquez sur le bouton ⏰ à côté du champ *Edit schema*.

5. Laissez la case *Set file location* décochée pour utiliser le chemin par défaut pour la table Hive.

6. Cochez la case *Set Delimited row format* pour afficher les options disponibles du format de ligne.

7. Cochez la case *Field* et saisissez un point-virgule (,) comme séparateur de champs dans le champ qui apparaît.

8. Cochez la case *Line* et laissez la valeur par défaut comme séparateur de lignes.

**Écrire des données dans la table**

**Pourquoi et quand exécuter cette tâche**

Configurer le *tHiveLoad*
Procédure

1. Double-cliquez sur le **tHiveLoad** pour ouvrir sa vue **Component**.

2. Cochez la case **Use an existing connection** et, dans la liste **Component list**, sélectionnez la connexion configurée dans le composant **tHiveConnection** utilisé dans ce Job.

3. Dans le champ **Load action**, sélectionnez **LOAD** pour écrire des données du fichier d’exemple présenté au début du scénario.

4. Dans le champ **File path**, saisissez le chemin d’accès au répertoire dans lequel sont stockées les données. Dans cet exemple, les données sont stockées sur le système HDFS utilisé.

Dans un cas d’utilisation réel, vous pouvez utiliser le **tHDFSOutput** afin d’écrire des données dans un système HDFS. Vous devez également vous assurer que l’application Hive a les droits et les permissions nécessaires pour écrire et également déplacer les données.

Pour plus d’informations sur le **tHDFSOutput**, consultez **tHDFSOutput** à la page 1620.

Pour plus d’informations sur les droits et permissions, consultez la documentation ou contactez l’administrateur du cluster Hadoop utilisé.

Notez que si vous devez lire des données depuis un système de fichiers local autre que le système HDFS, vous devez vous assurer que les données lues sont stockées dans le système de fichiers local de la machine sur laquelle le Job est exécuté. Cochez ensuite la case **Local** dans la vue **Basic settings**. Par exemple, lorsque le mode de connexion à Hive est **Standalone**, le Job est exécuté sur la machine sur laquelle l’application Hive est installée. Les données sont donc stockées sur cette machine.

5. Dans le champ **Table name**, saisissez le nom de la table cible dans laquelle charger les données. Dans ce scénario, saisissez **employees**.

6. Dans la liste **Action on file**, sélectionnez **APPEND**.

7. Cochez la case **Set partitions** et, dans le champ qui apparaît, saisissez le nom de la partition dans laquelle vous souhaitez ajouter des données. Dans ce scénario, cette partition est **country='US'**.

Configurer le **tHiveLoad**

Procédure

1. Double-cliquez sur le **tHiveLoad** pour ouvrir sa vue **Component**.
2. Cochez la case **Use an existing connection** et, dans la liste **Component list**, sélectionnez la connexion configurée dans le composant **tHiveConnection** utilisé dans ce Job.

3. Dans le champ **Load action**, sélectionnez **LOAD** pour écrire des données du fichier d’exemple présenté au début du scénario.

4. Dans le champ **File path**, saisissez le chemin d’accès au répertoire dans lequel sont stockées les données. Dans cet exemple, les données sont stockées sur le système HDFS utilisé. Dans un cas d’utilisation réel, vous pouvez utiliser le **tHDFSOutput** afin d’écrire des données dans un système HDFS. Vous devez également vous assurer que l’application Hive a les droits et les permissions nécessaires pour écrire et également déplacer les données.

   Pour plus d’informations sur les droits et permissions, consultez la documentation ou contactez l’administrateur du cluster Hadoop utilisé.

   Notez que si vous devez lire des données depuis un système de fichiers local autre que le système HDFS, vous devez vous assurer que les données lues sont stockées dans le système de fichiers local de la machine sur laquelle le Job est exécuté. Cochez ensuite la case **Local** dans la vue **Basic settings**. Par exemple, lorsque le mode de connexion à Hive est **Standalone**, le Job est exécuté sur la machine sur laquelle l’application Hive est installée. Les données sont donc stockées sur cette machine.

5. Dans le champ **Table name**, saisissez le nom de la table cible dans laquelle charger les données. Dans ce scénario, saisissez **employees**.

6. Dans la liste **Action on file**, sélectionnez **APPEND**.

7. Cochez la case **Set partitions** et, dans le champ qui apparaît, saisissez le nom de la partition dans laquelle vous souhaitez ajouter des données. Dans ce scénario, cette partition est **country='US'**.

### Exécuter le Job

Vous pouvez appuyer sur la touche **F6** pour exécuter ce Job.

Cela fait, la vue **Run** s’ouvre automatiquement. Vous pouvez suivre le processus d’exécution.

Vous pouvez également vérifier les résultats dans la console Web de votre distribution Hadoop.
Si vous souhaitez plus d’informations concernant le Job, il est recommandé d’utiliser la console Web du JobTracker fournie par votre distribution Hadoop.
tHiveCreateTable

Ce composant crée des tables Hive convenant à un grand nombre de formats de données Hive.

Un format de données Hive, comme RC ou ORC, vous permet d’obtenir une meilleure performance dans le traitement des données avec Hive.

Le tHiveCreateTable se connecte à la base de données Hive à utiliser et crée une table Hive dédiée aux données du format spécifié.

Propriétés du tHiveCreateTable Standard

Ces propriétés sont utilisées pour configurer le tHiveCreateTable s’exécutant dans le framework de Jobs Standard.

Le composant tHiveCreateTable Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

Configuration de la connexion :

- Lorsque vous utilisez ce composant avec Google Dataproc :

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project identifier</td>
<td>Saisissez l’ID de votre projet Google Cloud Platform.</td>
</tr>
<tr>
<td></td>
<td>Si vous n’êtes pas certain de l’ID de votre projet, vérifiez dans la page</td>
</tr>
<tr>
<td></td>
<td>Manage Resources de vos services Google Cloud Platform.</td>
</tr>
<tr>
<td>Cluster identifier</td>
<td>Saisissez l’ID de votre cluster Dataproc à utiliser.</td>
</tr>
<tr>
<td>Region</td>
<td>Saisissez les régions dans lesquelles sont utilisées les ressources de</td>
</tr>
<tr>
<td></td>
<td>calcul et dans lesquelles sont stockées et traitées les données. Si vous</td>
</tr>
<tr>
<td></td>
<td>n’avez pas besoin de spécifier une région en particulier, laissez la valeur</td>
</tr>
<tr>
<td></td>
<td>par défaut global.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations relatives aux régions disponibles et aux zones</td>
</tr>
<tr>
<td></td>
<td>de chaque groupe de région, consultez Regions and Zones (en anglais).</td>
</tr>
<tr>
<td>Google Storage staging bucket</td>
<td>Comme un Job Talend nécessite ses fichiers .jar dépendants pour être</td>
</tr>
<tr>
<td></td>
<td>exécuté, spécifiez le répertoire Google Storage dans lequel ces fichiers</td>
</tr>
<tr>
<td></td>
<td>.jar sont transférés afin que votre Job accède à ces fichiers lors de l’</td>
</tr>
<tr>
<td></td>
<td>exécution. Le répertoire à saisir doit se terminer par une barre oblique</td>
</tr>
<tr>
<td></td>
<td>(/). Si le répertoire n’existe pas, un répertoire est créé à la volée mais</td>
</tr>
<tr>
<td></td>
<td>le bucket à utiliser doit déjà exister.</td>
</tr>
<tr>
<td>Database</td>
<td>Renseignez ce champ avec le nom de la base de données.</td>
</tr>
</tbody>
</table>
| Provide Google Credentials in file | Lorsque vous lancez votre Job à partir d’une machine donnée sur laquelle Google Cloud SDK a été installé et vous a autorisé à utiliser vos identifiants de compte utilisateur pour accéder à Google Cloud Platform, ne cochez pas cette case. Dans cette situation, cette machine est souvent votre machine locale.
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Lorsque vous utilisez ce composant avec HDInsight :</td>
<td></td>
</tr>
</tbody>
</table>
| **WebHCat configuration** | Saisissez l’adresse et les informations d’authentification du cluster Microsoft HD Insight à utiliser. Par exemple, l’adresse peut être nom_de_votre_cluster_hdinsight.azurehdinsight.net et les informations d’authentification peuvent être votre nom de compte Azure : ychen. Le studio utilise ce service pour soumettre le Job au cluster HD Insight.
Dans le champ **Job result folder**, saisissez l’emplacement où vous souhaitez stocker les résultats d’exécution du Job dans Azure Storage. |
| **HDInsight configuration** | Saisissez les informations d’authentification du cluster HD Insight à utiliser. |
Dans le champ **Container**, saisissez le nom du conteneur à utiliser.
Dans le champ **Deployment Blob**, saisissez l’emplacement où vous souhaitez stocker le Job et ses bibliothèques dépendantes dans le compte Azure Storage. |
| **Database** | Renseignez ce champ avec le nom de la base de données. |
| • Lorsque vous utilisez les autres distributions : | |
| **Connection mode** | Sélectionnez un mode de connexion dans la liste. Les options varient en fonction de la distribution que vous utilisez. |
| **Hive server** | Sélectionnez le serveur Hive sur lequel vous souhaitez que le Job utilisant ce composant exécute des requêtes dans Hive.
La liste **Hive server** est disponible uniquement lorsque la distribution Hadoop à utiliser, par exemple HortonWorks Data Platform V1.2.0 (Bimota) supporte |
HiveServer2. Vous pouvez sélectionner HiveServer2 (**Hive 2**), le serveur supportant mieux les connexions simultanées de différents clients que HiveServer (**Hive 1**).


<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de la base de données.</td>
</tr>
</tbody>
</table>
| **Database** | Renseignez ce champ avec le nom de la base de données.  

**Remarque :**  
Ce champ n’est pas disponible lorsque vous sélectionnez l’option **Embedded** dans la liste **Connection mode**. |
| **Username et Password** | Données d’authentification de l’utilisateur de la base de données.  
Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Use kerberos authentication** | Si vous accédez au Metastore de Hive avec une sécurité Kerberos, cochez cette case et saisissez ensuite les paramètres appropriés dans les champs qui s’affichent.  
- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans **Connexion sécurisée à MapR** à la page 1745.  
  Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication et Use Kerberos authentication**.  
  MapR devrait pouvoir trouver automatiquement ce ticket à la volée.  

Les valeurs des paramètres suivants peuvent être trouvées dans le fichier *hive-site.xml* du système Hive utilisé.  

1. **Hive principal** utilise la valeur de *hive.metastore.kerberos.principal*. C’est le principal du service du Metastore de Hive.  
2. **HiveServer2 local user principal** utilise la valeur de *hive.server2.authentication.kerberos.principal*. |
3. **HiveServer2 local user keytab** utilise la valeur de `hive.server2.authentication.kerberos.keytab`.

4. **Metastore URL** utilise la valeur de `javax.jdo.option.ConnectionURL`. C'est la chaîne JDBC de connexion au Metastore de Hive.

5. **Driver class** utilise la valeur de `javax.jdo.option.ConnectionDriverName`. C'est le nom du pilote de la connexion JDBC.

6. **Username** utilise la valeur de `javax.jdo.option.ConnectionUserName`. Ce paramètre, ainsi que le paramètre **Password**, sont utilisés pour les informations de connexion de l’utilisateur au Metastore de Hive.

7. **Password** utilise la valeur de `javax.jdo.option.ConnectionPassword`.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

---

### Use a keytab to authenticate

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est `user1` et le principal à utiliser est `guest`. Dans cette situation, assurez-vous que `user1` a les droits de lecture pour le fichier keytab à utiliser.

---

### Use SSL encryption

Cochez cette case pour activer la connexion chiffrée SSL ou TLS.

Les champs qui s'affichent ensuite fournissent les informations d'authentification :

- Dans le champ **Trust store path**, saisissez le chemin ou parcourez votre système jusqu’au fichier TrustStore à utiliser. Par défaut, les types TrustStore supportés sont **JKS** et **PKCS 12**.

- Pour saisir le mot de passe, cliquez sur le bouton `[...]` à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Cette fonctionnalité n’est disponible que pour HiveServer2 en mode **Standalone** pour les distributions suivantes :

- Hortonworks Data Platform 2.0 +
Cochez cette case et, dans le champ qui s’affiche, saisissez l’emplacement du ResourceManager de votre distribution. Par exemple tal-qa114.tale nd.lan:8050.

Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d’un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du de l’exécution) :

1. Cochez la case **Set resourcemanager scheduler address** et saisissez l’adresse de l’ordonnanceur (Scheduler) dans le champ qui apparaît.

2. Cochez la case **Set jobhistory address** et saisissez l’emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory.

3. Cochez la case **Set staging directory** et saisissez le chemin d’accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l’exécution de programmes. Ce répertoire se trouve sous la propriété `yarn.app.mapreduce.am.staging-dir` dans les fichiers de configuration, notamment les fichiers `yarn-site.xml` et `mapred-site.xml` de votre distribution.

4. Allouez des volumes de mémoire aux calculs **Map** et **Reduce** et au service **ApplicationMaster** de YARN en cochant la case **Set memory** dans la vue **Advanced settings**.

5. Cochez la case **Set Hadoop user** et saisissez le nom de l’utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d’écriture, ce champ vous permet d’exécuter le Job directement avec l’utilisateur ayant les droits d’accès appropriés au fichier ou répertoire à traiter.

6. Cochez la case **Use datanode hostname** pour permettre au Job d’accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété `dfs.client.use.datanode.hostname` à `true`. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.

d’un système Hadoop. Par exemple, si vous avez choisi une machine nommée *masternode* en tant que NameNode, l’emplacement est `hdfs://masternode:portnumber`. Si vous utilisez WebHDFS, l’emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être `swebhdfs` et vous devez utiliser un `vLibraryLoad` dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.


Les autres propriétés :

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Use an existing connection**

| Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. **Au niveau parent**, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. **Au niveau enfant**, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.


**Distribution**

| Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique. |

- **Si disponible dans la liste de Distribution**, l’option **Microsoft HD Insight** vous permet d’utiliser un cluster Microsoft HD Insight. Dans cette optique, vous devez configurer les connexions au cluster HD.

- Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d'informations concernant la configuration d'un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L'option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d'index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l'évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d'utiliser l'option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l'opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d'Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d'importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez...
Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Hive version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
</table>
| Schema et Edit Schema        | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
  - **View schema** : sélectionnez cette option afin de voir le schéma.
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| Table Name                   | Nom de la table à créer.                                                                                                                                                           |
| Action on table              | Sélectionnez l’action à effectuer pour créer une table.                                                                                                                         |
| Format                       | Sélectionnez le format de données auquel la table à créer sera dédiée. Les formats de données disponibles varient selon la distribution Hadoop que vous utilisez. Notez que, lorsque le format de fichier à utiliser est PARQUET, il est possible qu’il vous soit demandé de |
trouver le fichier Jar Parquet spécifique et l’installer dans le studio.

- Lorsque le mode de connexion à Hive est **Embedded**, le Job est exécuté sur votre machine locale et appelle ce Jar installé dans le studio.

- Lorsque le mode de connexion à Hive est **Standalone**, le Job est exécuté sur le serveur hébergeant Hive et ce fichier Jar est envoyé au système HDFS du cluster auquel vous vous connectez. Assurez-vous d’avoir correctement défini l’URI du NameNode dans le champ correspondant de la vue **Basic settings**.


### Inputformat class et Outputformat class

| Ces champs apparaissent uniquement lorsque vous avez sélectionné **INPUTFORMAT and OUTPUTFORMAT** dans la liste **Format**. Ces champs vous permettent de saisir le nom des fichiers Jar à utiliser pour les formats de données non disponibles dans la liste **Format**. |

### Storage class

| Saisissez le nom du gestionnaire de stockage à utiliser pour créer une table non-native (table Hive stockée et gérée dans d’autres systèmes que Hive, par exemple Cassandra ou MongoDB). Ce champ est disponible uniquement lorsque vous avez sélectionné **STORAGE** dans la liste **Format**. Pour plus d’informations concernant les gestionnaires de stockage, consultez https://cwiki.apache.org/confluence/display/Hive/StorageHandlers (en anglais). |

### Set partitions

| Cochez cette case pour ajouter des colonnes de partition à la table à créer. Une fois cochée, vous devez définir le schéma des colonnes de partition à ajouter. |

### Set file location

| Si vous souhaitez créer une table Hive dans un répertoire différent de celui par défaut, cochez cette case et saisissez le chemin d’accès au répertoire HDFS à utiliser pour le contenu de la table. Cela est utile lorsque vous devez créer une table Hive externe en cochant la case **Create an external table** dans l’onglet **Advanced settings**. |

### Use S3 endpoint

| La case **Use S3 endpoint** apparaît lorsque vous cochez la case **Set file location** pour créer une table Hive externe. Après avoir coché la case **Use S3 endpoint**, vous devez renseigner les paramètres suivants dans les champs qui apparaissent :  
  - **S3 bucket** : saisissez le nom du bucket dans lequel vous souhaitez créer la table. |
- **Access key et Secret key** : renseignez les informations d’authentification requises pour vous connecter au bucket Amazon S3 à utiliser.

  Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Notez que le format du fichier S3 est S3N (S3 Native Filesystem).

Étant donné que la table Hive créée dans S3 est une table externe, la case **Use S3 endpoint** ne peut être utilisée qu’en cochant la case **Create an external table**.

### Advanced settings

<p>| <strong>Like table</strong> | Cochez cette case et saisissez le nom de la table Hive à copier. Cela vous permet de copier la définition d’une table existante sans copier ses données. Pour plus d’informations concernant le paramètre Like, consultez la documentation Apache relative au langage de définition des données Hive. |
| <strong>Create an external table</strong> | Cochez cette case afin de faire de la table créée une table Hive externe. Ce type de table Hive laisse les données brutes où elles sont si les données sont dans HDFS. Une table externe est habituellement le meilleur choix pour accéder aux données partagées existant dans un système de fichiers. Pour plus d’informations concernant une table externe Hive, consultez la documentation Apache relative à Hive. |
| <strong>Table comment</strong> | Saisissez la description à utiliser pour la table à créer. |
| <strong>As select</strong> | Cochez cette case et saisissez l'instruction As select pour créer une table Hive basée sur une instruction Select. |
| <strong>Set clustered_by or skewed_by statement</strong> | Saisissez l'instruction Clustered by pour mettre en cluster les données d’une table ou diviser une partition en buckets, et/ou saisissez l'instruction Skewed by pour autoriser Hive à séparer les données très asymétriques et les mettre dans des fichiers séparés. Cela est généralement utilisé pour obtenir de meilleures performances durant les requêtes. |
| <strong>SerDe properties</strong> | Si vous utilisez le format de lignes SerDe, vous pouvez ajouter des propriétés SerDe personnalisées pour écraser les propriétés par défaut utilisées par le moteur Hadoop du studio. |</p>
<table>
<thead>
<tr>
<th>Table properties</th>
<th>Ajoutez une table de propriétés Hive personnalisée pour écraser les propriétés utilisées par défaut par le moteur Hadoop du studio.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporary path</td>
<td>Si vous ne souhaitez pas configurer le Jobtracker et le NameNode lorsque vous exécutez la requête <code>select * from your_table_name</code>, vous devez paramétrer un chemin d'accès temporaire. Par exemple, <code>/C:/select_all</code> sous Windows.</td>
</tr>
</tbody>
</table>
| Hadoop properties | Le *Studio Talend* utilise une configuration par défaut pour son moteur, afin d'effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l'exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.  
  
  - Notez que, si vous utilisez les métadonnées stockées centralement dans le *Repository*, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste *Property type*, vous passiez de *Repository* à *Built-in*.  
  
  Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :  
  
  
| Hive properties  | Le *Studio Talend* utilise la configuration par défaut pour son moteur afin d’effectuer des opérations dans un base de données Hive. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez cette table avec la (les) propriété(s) à personnaliser. Ensuite, à l’exécution, la (les) propriété(s) personnalisée(s) écrasent celles par défaut. Pour plus d’informations concernant les propriétés dédiées à Hive, consultez [https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration](https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration) (en anglais).  
  
  - Si vous devez utiliser Tez pour exécuter votre Job Hive, ajoutez `hive.execution.engine` à la colonne |
Properties et Tez à la valeur Value, en entourant ces chaînes de caractères de guillemets doubles.

- Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste Property type, vous passiez de Repository à Built-in.

| Mapred job map memory mb et Mapred job reduce memory mb | Vous pouvez personnaliser les opérations map et reduce en cochant la case Set memory, pour configurer les allocations de mémoire pour ces opérations à effectuer par le système Hadoop.
Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations map et reduce dans les champs Mapred job map memory mb et Mapred job reduce memory mb, respectivement. Par défaut, les valeurs sont toutes les deux 1000, ce qui est normalement adapté pour l’exécution de ces opérations. |
| Path separator in server | Laissez le champ Path separator in server tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d’autres termes, changez le séparateur si celui-ci n’est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte. |
| tStatCatcher Statistics | Cochez cette case afin de collecter les données de log au niveau des composants. |

### Global Variables

| Global Variables | QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères. 
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. 
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. 
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. 
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant fonctionne en standalonne. |
Si le Studio Talend utilisé pour vous connecter à une base de données Hive fonctionne sous Windows, vous devez créer manuellement un dossier appelé *tmp* à la racine du disque où le Studio Talend est installé.

### Row format

<table>
<thead>
<tr>
<th><strong>Set delimited row format</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set SerDe row format</strong></td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
</tr>
</tbody>
</table>

### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La tableDynamic settings est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue **Basic settings** devient inutilisable.


### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).
Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]** dans le menu **Window**. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

### Scénario associé

Pour un scénario associé, consultez **Scénario : Créer une table Hive partitionnée** à la page 1680.
**tHiveInput**

Ce composant extrait des données de Hive et les envoie au composant qui suit.

Le tHiveInput est le composant dédié à la base de données Hive (le système de data warehouse Hive). Il peut exécuter une requête HiveQL données afin d’extraire des données de Hive.

**Propriétés du tHiveInput Standard**

Ces propriétés sont utilisées pour configurer le tHiveInput s’exécutant dans le framework de Jobs Standard.

Le composant tHiveInput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

Configuration de la connexion :

- Lorsque vous utilisez ce composant avec Google Dataproc :

  | **Project identifier** | Saisissez l’ID de votre projet Google Cloud Platform. Si vous n’êtes pas certain de l’ID de votre projet, vérifiez dans la page Manage Resources de vos services Google Cloud Platform. |
  | **Cluster identifier** | Saisissez l’ID de votre cluster Dataproc à utiliser. |
  | **Region** | Saisissez les régions dans lesquelles sont utilisées les ressources de calcul et dans lesquelles sont stockées et traitées les données. Si vous n’avez pas besoin de spécifier une région en particulier, laissez la valeur par défaut *global*.

  Pour plus d’informations relatives aux régions disponibles et aux zones de chaque groupe de région, consultez *Regions and Zones* (en anglais).

  | **Google Storage staging bucket** | Comme un Job Talend nécessite ses fichiers .jar dépendants pour être exécuté, spécifiez le répertoire Google Storage dans lequel ces fichiers .jar sont transférés afin que votre Job accède à ces fichiers lors de l’exécution.

  Le répertoire à saisir doit se terminer par une barre oblique (/). Si le répertoire n’existe pas, un répertoire est créé à la volée mais le bucket à utiliser doit déjà exister. |
  | **Database** | Renseignez ce champ avec le nom de la base de données. |
### Access Key et Secret Key
Saisissez les informations d'authentification obtenues de Google pour que le composant tHiveInput lise les données temporaires de Google Storage.

Ces clés peuvent être consultées dans la vue Interoperable Access, sous l'onglet Google Cloud Storage du projet dans la console d'API de Google.

Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ *Secret key*, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.


### Provide Google Credentials in file
Lorsque vous lancez votre Job à partir d’une machine donnée sur laquelle Google Cloud SDK a été installé et vous a autorisé à utiliser vos identifiants de compte utilisateur pour accéder à Google Cloud Platform, ne cochez pas cette case. Dans cette situation, cette machine est souvent votre machine locale.


- Lorsque vous utilisez ce composant avec HDInsight :

#### WebHCat configuration

Dans le champ *Job result folder*, saisissez l'emplacement où vous souhaitez stocker les résultats d'exécution du Job dans Azure Storage.

#### HDInsight configuration
Saisissez les informations d'authentification du cluster HD Insight à utiliser.

#### Windows Azure Storage configuration

Dans le champ *Container*, saisissez le nom du conteneur à utiliser.

Dans le champ *Deployment Blob*, saisissez l'emplacement où vous souhaitez stocker le Job et
ses bibliothèques dépendantes dans le compte Azure Storage.

<table>
<thead>
<tr>
<th><strong>Database</strong></th>
<th>Renseignez ce champ avec le nom de la base de données.</th>
</tr>
</thead>
</table>

- Lorsque vous utilisez les autres distributions :

<table>
<thead>
<tr>
<th><strong>Connection mode</strong></th>
<th>Sélectionnez un mode de connexion dans la liste. Les options varient en fonction de la distribution que vous utilisez.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Hive server</strong></th>
<th>Sélectionnez le serveur Hive sur lequel vous souhaitez que le Job utilisant ce composant exécute des requêtes dans Hive.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>La liste Hive server est disponible uniquement lorsque la distribution Hadoop à utiliser, par exemple HortonWorks Data Platform V1.2.0 (Bimota) supporte HiveServer2. Vous pouvez sélectionner HiveServer2 (Hive 2), le serveur supportant mieux les connexions simultanées de différents clients que HiveServer (Hive 1).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de la base de données.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Port</strong></th>
<th>Numéro du port d’écoute du serveur de la base de données.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Database</strong></th>
<th>Renseignez ce champ avec le nom de la base de données.</th>
</tr>
</thead>
</table>

**Remarque :**

Ce champ n’est pas disponible lorsque vous sélectionnez l’option Embedded dans la liste Connection mode.

<table>
<thead>
<tr>
<th><strong>Username et Password</strong></th>
<th>Données d’authentification de l’utilisateur de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Use kerberos authentication</strong></th>
<th>Si vous accédez au Metastore de Hive avec une sécurité Kerberos, cochez cette case et saisissez ensuite les paramètres appropriés dans les champs qui s’affichent.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.</td>
</tr>
</tbody>
</table>
Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Les valeurs des paramètres suivants peuvent être trouvées dans le fichier *hive-site.xml* du système Hive utilisé.

1. **Hive principal** utilise la valeur de `hive.metastore.kerberos.principal`. C’est le principal du service du Metastore de Hive.
2. **HiveServer2 local user principal** utilise la valeur de `hive.server2.authentication.kerberos.principal`.
3. **HiveServer2 local user keytab** utilise la valeur de `hive.server2.authentication.kerberos.keytab`.
4. **Metastore URL** utilise la valeur de `javax.jdo.option.ConnectionURL`. C’est la chaîne JDBC de connexion au Metastore de Hive.
5. **Driver class** utilise la valeur de `javax.jdo.option.ConnectionDriverName`. C’est le nom du pilote de la connexion JDBC.
6. **Username** utilise la valeur de `javax.jdo.option.ConnectionUserName`. Ce paramètre, ainsi que le paramètre **Password**, sont utilisés pour les informations de connexion de l’utilisateur au Metastore de Hive.
7. **Password** utilise la valeur de `javax.jdo.option.ConnectionPassword`.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

---

**Use a keytab to authenticate**

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser.
| Use SSL encryption | Cochez cette case pour activer la connexion chiffrée SSL ou TLS.

Les champs qui s’affichent ensuite fournissent les informations d’authentification :

- Dans le champ Trust store path, saisissez le chemin ou parcourez votre système jusqu’au fichier TrustStore à utiliser. Par défaut, les types TrustStore supportés sont JKS et PKCS 12.
- Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

Cette fonctionnalité n’est disponible que pour HiveServer2 en mode Standalone pour les distributions suivantes :

- Hortonworks Data Platform 2.0 +
- Cloudera CDH4 +
- Pivotal HD 2.0 +
- Amazon EMR 4.0.0 +


Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d’un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du de l’exécution) :

1. Cochez la case Set resourcemanager scheduler address et saisissez l’adresse de l’ordonnanceur (Scheduler) dans le champ qui apparaît.


3. Cochez la case Set staging directory et saisissez le chemin d’accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l’exécution de programmes. Ce répertoire se trouve sous la propriété yarn.app.mapreduce.am.staging-dir dans les fichiers de configuration, notamment les fichiers yarn-site.xml et mapred-site.xml de votre distribution.

4. Allouez des volumes de mémoire aux calculs Map et Reduce et au service ApplicationMaster de YARN en cochant la case Set memory dans la vue Advanced settings.

5. Cochez la case Set Hadoop user et saisissez le nom de l’utilisateur avec lequel vous souhaitez
exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d’écriture, ce champ vous permet d’exécuter le Job directement avec l’utilisateur ayant les droits d’accès appropriés au fichier ou répertoire à traiter.

6. Cochez la case **Use datanode hostname** pour permettre au Job d’accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété `dfs.client.use.datanode.hostname` à `true`. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.


<table>
<thead>
<tr>
<th>Set NameNode URI</th>
</tr>
</thead>
</table>
| Cochez cette case et, dans le champ qui s’affiche, saisissez l’URI du NameNode Hadoop, le nœud maître d’un système Hadoop. Par exemple, si vous avez choisi une machine nommée `masternode` en tant que NameNode, l'emplacement est `hdfs://masternode:portnumber`. Si vous utilisez WebHDFS, l'emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être `swebhdfs` et vous devez utiliser un `TLibraryLoad` dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.  

Les autres propriétés :

<table>
<thead>
<tr>
<th>Property type</th>
</tr>
</thead>
</table>
| Peut être **Built-In** ou **Repository**.  
**Built-In** : propriétés utilisées ponctuellement.  
**Repository** : sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |

<table>
<thead>
<tr>
<th>Use an existing connection</th>
</tr>
</thead>
</table>
| Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
**Remarque** :  
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :  
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion. |
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.</td>
</tr>
<tr>
<td>• Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
<tr>
<td>• L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.</td>
</tr>
</tbody>
</table>

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous
connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Hive version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
</table>
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

<table>
<thead>
<tr>
<th>Built-in</th>
<th>Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</th>
</tr>
</thead>
</table>

<p>| Table Name | Nom de la table à traiter. |</p>
<table>
<thead>
<tr>
<th>Query type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.</td>
</tr>
</tbody>
</table>

**Guess Query**

Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ Query.

**Guess schema**

Cliquez sur ce bouton afin de récupérer le schéma de la table.

**This query uses Parquet objects**

Lorsqu'elle est disponible, cochez cette case pour indiquer que la table à gérer utilise le format Parquet et permet au composant d'appeler le fichier Jar requis.

Notez que, lorsque le format de fichier à utiliser est **PARQUET**, il est possible qu'il vous soit demandé de trouver le fichier Jar Parquet spécifique et l'installer dans le studio.

- Lorsque le mode de connexion à Hive est **Embedded**, le Job est exécuté sur votre machine locale et appelle ce Jar installé dans le studio.
- Lorsque le mode de connexion à Hive est **Standalone**, le Job est exécuté sur le serveur hébergeant Hive et ce fichier Jar est envoyé au système HDFS du cluster auquel vous vous connectez. Assurez-vous d'avoir correctement défini l'URI du NameNode dans le champ correspondant de la vue **Basic settings**.

Ce fichier Jar est téléchargeable depuis le site Web d'Apache. Vous pouvez trouver plus d'informations concernant l'installation des modules externes dans Talend Help Center (https://help.talend.com).

**Query**

Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.

Pour plus d’informations concernant le langage de requêtes Hive, consultez https://cwiki.apache.org/confluence/display/Hive/LanguageManual (en anglais).

**Remarque :**


Hadoop fournit différents formats de compression permettant de réduire l’espace nécessaire au stockage des fichiers et d’accélérer le transfert de données. Lorsque vous lisez un fichier compressé, le **Studio**
**Execution engine**

Cochez cette case et, dans la liste déroulante, sélectionnez le framework à utiliser pour exécuter le Job.

Cette liste est disponible lorsque vous utilisez le mode **Embedded** pour la connexion et distribution Hive avec laquelle vous travaillez, parmi les suivantes :

- Hortonworks : V2.1 et V2.2.
- MapR : V4.0.1.
- **Custom** : cette option vous permet de vous connecter à une distribution supportant mais non officiellement supportée par **Talend**.

Avant d’utiliser Tez, vérifiez que votre cluster Hadoop supporte Tez. Vous devez configurer l’accès aux bibliothèques Tez correspondantes via la vue **Advanced settings** de ce composant.

Pour plus d’informations concernant Hive avec Tez, consultez la documentation Apache à l’adresse https://cwiki.apache.org/confluence/display/Hive/Hive+on+Tez (en anglais). Des exemples vous sont présentés afin d’expliquer comment Tez peut être utilisé pour optimiser les performances par rapport à MapReduce.

---

**Advanced settings**

**Tez lib**

Choisissez comment accéder aux bibliothèques de Tez :

- **Auto install** : lors de l’exécution, le Job charge et déploie les bibliothèques de Tez fournies par le Studio dans le répertoire spécifié dans le champ **Install folder in HDFS**, par exemple, `/tmp/usr/tez`.
  
  Si vous avez configuré la propriété `tez.lib.uris` dans la table des propriétés, ce répertoire écrase la valeur de la propriété lors de l’exécution. Les autres propriétés configurées dans la table sont toujours prises en compte.

- **Use exist** : le Job accède aux bibliothèques de Tez déjà déployées dans le cluster Hadoop à utiliser. Vous devez saisir le chemin d’accès pointant vers ces bibliothèques dans le champ **Lib path (folder or file)**.

- **Lib jar** : cette table s’affiche lorsque vous avez sélectionné **Auto install** dans la liste **Tez lib** et que vous utilisez une distribution personnalisée (Custom). Dans cette table, vous devez ajouter les bibliothèques de Tez à charger.

**Temporary path**

Si vous ne souhaitez pas configurer le Jobtracker et le NameNode lorsque vous exécutez la requête `select * from your_table_name`, vous devez paramétrer un chemin d’accès temporaire. Par exemple, `/C:/select_all` sous Windows.
<table>
<thead>
<tr>
<th>Trim all the String/Char columns</th>
<th>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim column</td>
<td>Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
</tbody>
</table>
|                                 | **Remarque :**  
|                                 | Décochez la case Trim all the String/Char columns pour activer l’option Trim column. |
| Hadoop properties               | Le Studio Talend utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.  
|                                 | • Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste **Property type**, vous passiez de Repository à Built-in.  
|                                 | Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :  
| Hive properties                 | Le Studio Talend utilise la configuration par défaut pour son moteur afin d’effectuer des opérations dans un base de données Hive. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez cette table avec la (les) propriété(s) à personnaliser. Ensuite, à l’exécution, la (les) propriété(s) personnalisée(s) écrasent celles par défaut. Pour plus d’informations concernant les propriétés dédiées à Hive, consultez [https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration](https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration) (en anglais). |
Si vous devez utiliser Tez pour exécuter votre Job Hive, ajoutez `hive.execution.engine` à la colonne `Properties` et `Tez` à la valeur `Value`, en entourant ces chaînes de caractères de guillemets doubles.

Notez que, si vous utilisez les métadonnées stockées centralement dans le `Repository`, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste `Property type`, vous passiez de `Repository` à `Built-in`.

**Mapred job map memory mb et Mapred job reduce memory mb**

Vous pouvez personnaliser les opérations map et reduce en cochant la case `Set memory`, pour configurer les allocations de mémoire pour ces opérations à effectuer par le système Hadoop.

Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations `map` et `reduce` dans les champs `Mapred job map memory mb` et `Mapred job reduce memory mb`, respectivement. Par défaut, les valeurs sont toutes les deux 1000, ce qui est normalement adapté pour l’exécution de ces opérations.

Les paramètres de mémoire à définir sont `Map (in Mb)`, `Reduce (in Mb)` et `ApplicationMaster (in Mb)`. Ces champs permettent d’allouer dynamiquement de la mémoire aux opérations map et reduce et à l’ApplicationMaster de YARN.

**Path separator in server**

Laissez le champ `Path separator in server` tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d’autres termes, changez le séparateur si celui-ci n’est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

- `NB_LINE` : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable `After` et retourne un entier.
- `QUERY` : requête traitée. Cette variable est une variable `Flow` et retourne une chaîne de caractères.
- `ERROR_MESSAGE` : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case `Die on error` est décochée, si le composant a cette option.

Une variable `Flow` fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace`
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d'utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes Hive QL.
Si le Studio Talend utilisé pour vous connecter à une base de données Hive fonctionne sous Windows, vous devez créer manuellement un dossier appelé tmp à la racine du disque où le Studio Talend est installé. |
|---------------------|--------------------------------------------------------------------------------------------------|

#### HBase Configuration

| Remarque : Disponible uniquement lorsque la case Use an existing connection est décochée. |
| Store by HBase |
| Zookeeper quorum |
| Zookeeper client port |
| Define the jars to register for HBase |
| Register jar for HBase |

#### Dynamic settings

| Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend. |
| La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. |
| Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les |
Pour un scénario utilisant un composant d’entrée dans un Job, consultez Scénario : Ecrire des colonnes d’une base de données MySQL dans un fichier de sortie en utilisant tMySqlInput à la page 2635 du composant tMysqlInput.

Lors de la configuration de ce composant, gardez à l’esprit que des paramètres sont requis par Hadoop, tels que le NameNode et le Jobtracker, puisque le composant doit se connecter à une distribution Hadoop.
**tHiveLoad**

Ce composant écrit des données de différents formats dans une table Hive donnée ou pour exporter des données d’une table Hive vers un répertoire.

Le tHiveLoad se connecte à une base de données Hive et copie ou déplace des données dans une table Hive existante ou dans un répertoire spécifié.

**Propriétés du tHiveLoad Standard**

Ces propriétés sont utilisées pour configurer le tHiveLoad s’exécutant dans le framework de Jobs Standard.

Le composant tHiveLoad Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

Configuration de la connexion :

- Lorsque vous utilisez ce composant avec Google Dataproc :

  | **Project identifier** | Saisissez l’ID de votre projet Google Cloud Platform. Si vous n’êtes pas certain de l’ID de votre projet, vérifiez dans la page Manage Resources de vos services Google Cloud Platform. |
  | **Cluster identifier** | Saisissez l’ID de votre cluster Dataproc à utiliser. |
  | **Region** | Saisissez les régions dans lesquelles sont utilisées les ressources de calcul et dans lesquelles sont stockées et traitées les données. Si vous n’avez pas besoin de spécifier une région en particulier, laissez la valeur par défaut global. Pour plus d’informations relatives aux régions disponibles et aux zones de chaque groupe de région, consultez Regions and Zones (en anglais). |
  | **Google Storage staging bucket** | Comme un Job Talend nécessite ses fichiers .jar dépendants pour être exécuté, spécifiez le répertoire Google Storage dans lequel ces fichiers .jar sont transférés afin que votre Job accède à ces fichiers lors de l’exécution. Le répertoire à saisir doit se terminer par une barre oblique (/). Si le répertoire n’existe pas, un répertoire est créé à la volée mais le bucket à utiliser doit déjà exister. |
  | **Database** | Renseignez ce champ avec le nom de la base de données. |
  | **Provide Google Credentials in file** | Lorsque vous lancez votre Job à partir d’une machine donnée sur laquelle Google Cloud SDK a été installé |
et vous a autorisé à utiliser vos identifiants de compte utilisateur pour accéder à Google Cloud Platform, ne cochez pas cette case. Dans cette situation, cette machine est souvent votre machine locale.


• Lorsque vous utilisez ce composant avec HDInsight :


Dans le champ Job result folder, saisissez l’emplacement où vous souhaitez stocker les résultats d’exécution du Job dans Azure Storage. |
|-----------------------|------------------------------------------------------------------------------------------------|

<table>
<thead>
<tr>
<th>HDInsight configuration</th>
<th>Saisissez les informations d’authentification du cluster HD Insight à utiliser.</th>
</tr>
</thead>
</table>


Dans le champ Container, saisissez le nom du conteneur à utiliser.

Dans le champ Deployment Blob, saisissez l’emplacement où vous souhaitez stocker le Job et ses bibliothèques dépendantes dans le compte Azure Storage. |
|----------------------|--------------------------------------------------------------------------------------------------|

<table>
<thead>
<tr>
<th>Database</th>
<th>Renseignez ce champ avec le nom de la base de données.</th>
</tr>
</thead>
</table>

• Lorsque vous utilisez les autres distributions :

<table>
<thead>
<tr>
<th>Connection mode</th>
<th>Sélectionnez un mode de connexion dans la liste. Les options varient en fonction de la distribution que vous utilisez.</th>
</tr>
</thead>
</table>

| Hive server | Sélectionnez le serveur Hive sur lequel vous souhaitez que le Job utilisant ce composant exécute des requêtes dans Hive.

La liste Hive server est disponible uniquement lorsque la distribution Hadoop à utiliser, par exemple HortonWorks Data Platform V1.2.0 (Bimota) supporte HiveServer2. Vous pouvez sélectionner HiveServer2 (Hive 2), le serveur supportant mieux les connexions |
simultanées de différents clients que HiveServer (Hive 1).


<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Renseignez ce champ avec le nom de la base de données.</td>
</tr>
</tbody>
</table>

Remarque :
Ce champ n'est pas disponible lorsque vous sélectionnez l’option **Embedded** dans la liste **Connection mode**.

<table>
<thead>
<tr>
<th><strong>Username et Password</strong></th>
<th>Données d’authentification de l’utilisateur de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Use kerberos authentication</strong></th>
<th>Si vous accédez au Metastore de Hive avec une sécurité Kerberos, cochez cette case et saisissez ensuite les paramètres appropriés dans les champs qui s’affichent.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans <strong>Connexion sécurisée à MapR</strong> à la page 1745.</td>
</tr>
<tr>
<td></td>
<td>Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases <strong>Force MapR ticket authentication et Use Kerberos authentication</strong>. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.</td>
</tr>
<tr>
<td></td>
<td>Les valeurs des paramètres suivants peuvent être trouvées dans le fichier <strong>hive-site.xml</strong> du système Hive utilisé.</td>
</tr>
<tr>
<td></td>
<td>1. <strong>Hive principal</strong> utilise la valeur de <strong>hive.metastore.kerberos.principal</strong>. C’est le principal du service du Metastore de Hive.</td>
</tr>
<tr>
<td></td>
<td>2. <strong>HiveServer2 local user principal</strong> utilise la valeur de <strong>hive.server2.authentication.kerberos.principal</strong>.</td>
</tr>
</tbody>
</table>
3. **HiveServer2 local user keytab** utilise la valeur de `hive.server2.authentication.kerberos.keytab`.

4. **Metastore URL** utilise la valeur de `javax.jdo.option.ConnectionURL`. C'est la chaîne JDBC de connexion au Metastore de Hive.

5. **Driver class** utilise la valeur de `javax.jdo.option.ConnectionDriverName`. C'est le nom du pilote de la connexion JDBC.

6. **Username** utilise la valeur de `javax.jdo.option.ConnectionUserName`. Ce paramètre, ainsi que le paramètre **Password**, sont utilisés pour les informations de connexion de l'utilisateur au Metastore de Hive.

7. **Password** utilise la valeur de `javax.jdo.option.ConnectionPassword`. Cette case est disponible ou indisponible selon la distribution d'Hadoop à laquelle vous vous connectez.

---

### Use a keytab to authenticate

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l'aide d'un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est **user1** et le principal à utiliser est **guest**. Dans cette situation, assurez-vous que **user1** a les droits de lecture pour le fichier keytab à utiliser.

### Use SSL encryption

Cochez cette case pour activer la connexion chiffrée SSL ou TLS.

Les champs qui s'affichent ensuite fournissent les informations d'authentification :

- Dans le champ **Trust store path**, saisissez le chemin ou parcourez votre système jusqu'au fichier TrustStore à utiliser. Par défaut, les types TrustStore supportés sont **JKS** et **PKCS 12**.

- Pour saisir le mot de passe, cliquez sur le bouton `[...]` à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Cette fonctionnalité n'est disponible que pour HiveServer2 en mode **Standalone** pour les distributions suivantes :

- Hortonworks Data Platform 2.0 +
- Cloudera CDH4 +
- Pivotal HD 2.0 +
- Amazon EMR 4.0.0 +

### Set Resource Manager

Cochez cette case et, dans le champ qui s’affiche, saisissez l’emplacement du ResourceManager de votre distribution. Par exemple `tal-qall4.tale nd.1an:8050`.

Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d’un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du de l’exécution) :

1. Cochez la case **Set resourcemanager scheduler address** et saisissez l’adresse de l’ordonnanceur (Scheduler) dans le champ qui apparaît.

2. Cochez la case **Set jobhistory address** et saisissez l’emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory.

3. Cochez la case **Set staging directory** et saisissez le chemin d’accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l’exécution de programmes. Ce répertoire se trouve sous la propriété `yarn.app.mapreduce.am.staging-dir` dans les fichiers de configuration, notamment les fichiers `yarn-site.xml` et `mapred-site.xml` de votre distribution.

4. Allouez des volumes de mémoire aux calculs Map et Reduce et au service ApplicationMaster de YARN en cochant la case **Set memory** dans la vue **Advanced settings**.

5. Cochez la case **Set Hadoop user** et saisissez le nom de l’utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d’écriture, ce champ vous permet d’exécuter le Job directement avec l’utilisateur ayant les droits d’accès appropriés au fichier ou répertoire à traiter.

6. Cochez la case **Use datanode hostname** pour permettre au Job d’accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété `dfs.client.use.datanode.hostname` à `true`. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.


### Set NameNode URI

Cochez cette case et, dans le champ qui s’affiche, saisissez l’URI du NameNode Hadoop, le nœud maître.
d’un système Hadoop. Par exemple, si vous avez choisi une machine nommée *masternode* en tant que NameNode, l’emplacement est `hdfs://masternode:portnumber`. Si vous utilisez WebHDFS, l’emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être `swebhdfs` et vous devez utiliser un `tLibraryLoad` dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.


### Les autres propriétés :

| Property type | Peut être **Built-In** ou **Repository**.  
**Built-In** : propriétés utilisées ponctuellement.  
**Repository** : sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |
|---------------|-------------------------------------------------------------------|

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adapté à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
**Remarque** :  
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :  
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.  
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.  
Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend. |
|-------------------------|-------------------------------------------------------------------|

| Distribution | Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
- Si disponible dans la liste de **Distribution**, l’option **Microsoft HD Insight** vous permet d’utiliser un cluster Microsoft HD Insight. Dans cette optique, vous devez configurer les connexions au cluster HD Insight et au service Windows Azure Storage du cluster dans les zones affichées. Pour plus d’informations |
|-----------------|-------------------------------------------------------------------|
concernant ces paramètres, recherchez Configurer manuellement la connexion, sur Talend Help Center (https://help.talend.com).

- Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d'informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d'Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez
**Connexion à une distribution Hadoop personnalisée à la page 1677.**

<table>
<thead>
<tr>
<th><strong>Hive version</strong></th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Load action</strong></td>
<td>Sélectionnez l’action à effectuer pour écrire des données à l’emplacement spécifié.</td>
</tr>
<tr>
<td></td>
<td>• Lorsque vous sélectionnez <strong>LOAD</strong>, cela permet de déplacer ou copier des données d’un répertoire spécifié.</td>
</tr>
<tr>
<td></td>
<td>• Lorsque vous sélectionnez <strong>INSERT</strong>, cela permet de déplacer ou copier des données à partir de requêtes.</td>
</tr>
<tr>
<td><strong>Execution engine</strong></td>
<td>Cochez cette case et, dans la liste déroulante, sélectionnez le framework à utiliser pour effectuer l’action <strong>INSERT</strong>.</td>
</tr>
<tr>
<td></td>
<td>Cette liste est disponible lorsque vous utilisez le mode <strong>Embedded</strong> pour la connexion et distribution Hive avec laquelle vous travaillez, parmi les suivantes :</td>
</tr>
<tr>
<td></td>
<td>• Hortonworks : V2.1 et V2.2.</td>
</tr>
<tr>
<td></td>
<td>• MapR : V4.0.1.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Custom</strong> : cette option vous permet de vous connecter à une distribution supportant mais non officiellement supportée par <strong>Talend</strong>.</td>
</tr>
<tr>
<td></td>
<td>Avant d’utiliser Tez, vérifiez que votre cluster Hadoop supporte Tez. Vous devez configurer l’accès aux bibliothèques Tez correspondantes via la vue <strong>Advanced settings</strong> de ce composant.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant Hive avec Tez, consultez la documentation Apache à l’adresse <a href="https://cwiki.apache.org/confluence/display/Hive/Hive+on+Tez">https://cwiki.apache.org/confluence/display/Hive/Hive+on+Tez</a> (en anglais). Des exemples vous sont présentés afin d’expliquer comment Tez peut être utilisé pour optimiser les performances par rapport à MapReduce.</td>
</tr>
<tr>
<td><strong>Target type</strong></td>
<td>Cette liste déroulante apparaît uniquement lorsque vous avez sélectionné <strong>INSERT</strong> dans la liste <strong>Load action</strong>.</td>
</tr>
<tr>
<td></td>
<td>Sélectionnez dans cette liste le type d’emplacement dans lequel écrire des données.</td>
</tr>
<tr>
<td></td>
<td>• Si vous sélectionnez <strong>Table</strong> comme destination, vous pouvez choisir d’écrire les données à la suite ou d’écaser le contenu dans la table spécifiée.</td>
</tr>
<tr>
<td></td>
<td>• Si vous sélectionnez <strong>Directory</strong> (répertoire) comme destination, le contenu de répertoire spécifié sera écrasé.</td>
</tr>
<tr>
<td><strong>Table name</strong></td>
<td>Saisissez le nom de la table Hive dans laquelle écrire les données.</td>
</tr>
</tbody>
</table>
Notez qu’avec l’action **INSERT**, ce champ est disponible uniquement lorsque vous avez sélectionné **Table** dans la liste **Target type**.

### File path

Saisissez le chemin d’accès au répertoire duquel lire les données ou dans lequel écrire des données, selon l’action sélectionnée dans la liste **Load action**.

- Si vous avez sélectionné **LOAD** : saisissez le chemin d’accès aux données à copier ou déplacer dans la table Hive spécifiée.
- Si vous avez sélectionné **INSERT** : saisissez le chemin d’accès au répertoire dans lequel exporter les données à partir d’une table Hive. Avec cette option, le champ **File path** est disponible uniquement lorsque vous avez sélectionné **Directory** dans la liste **Target type**.

### The target table uses the Parquet format

Si la table dans laquelle vous devez écrire les données est une table Parquet, cochez cette case.

Notez que, lorsque le format de fichier à utiliser est **PARQUET**, il est possible qu’il vous soit demandé de trouver le fichier Jar Parquet spécifique et l’installer dans le studio.

- Lorsque le mode de connexion à Hive est **Embedded**, le Job est exécuté sur votre machine locale et appelle ce Jar installé dans le studio.
- Lorsque le mode de connexion à Hive est **Standalone**, le Job est exécuté sur le serveur hébergeant Hive et ce fichier Jar est envoyé au système HDFS du cluster auquel vous vous connectez. Assurez-vous d’avoir correctement défini l’URI du NameNode dans le champ correspondant de la vue **Basic settings**.


Dans la liste **Compression** qui apparaît, sélectionnez le mode de compression à utiliser pour gérer le fichier Parquet. Le mode par défaut est **Uncompressed**.

### Action on file

Sélectionnez l’action à effectuer pour écrire les données. Cette liste est disponible uniquement lorsque la cible est une table Hive. Si la cible est un répertoire, l’action à effectuer est automatiquement **OVERWRITE**.

### Query

Ce champ apparaît uniquement lorsque vous avez sélectionné **INSERT** dans la liste **Load action**.

Saisissez la requête appropriée pour sélectionner les données à exporter dans la table ou le répertoire Hive spécifié(e).

### Local

Cochez cette case pour utiliser l’instruction Hive **LOCAL** pour accéder à un répertoire local. Notez que ce répertoire local est en fait la machine sur laquelle le
Job est exécuté. Ainsi, lorsque le mode de connexion à Hive sélectionné est Standalone, le Job s'exécute sur la machine sur laquelle l’application Hive est installée. Le dossier local se trouve alors sur cette machine.

Cette instruction est utilisée avec le répertoire défini dans le champ File path. La case Local est disponible uniquement lorsque le champ File path est disponible.

- Si vous sélectionnez l'action LOAD, le tHiveLoad copie les données locales dans la table cible.
- Si vous sélectionnez l'action INSERT, le tHiveLoad copie les données dans un répertoire local.
- Si vous laissez la case Local décochée, le répertoire défini dans le champ File path est supposé être dans le système HDFS à utiliser et les données sont déplacées à l’emplacement cible.

Pour plus d’informations concernant cette instruction LOCAL, consultez la documentation Apache concernant le langage Hive.

Set partitions
Cochez cette case pour utiliser la clause Hive Partition pour charger ou insérer des données dans une table Hive. Vous devez saisir les clés de partition et leur valeur à utiliser dans le champ qui apparaît.

Par exemple, saisissez `contry='US', state='CA'`. Cela permet de créer une clause de partition `Partition (contry='US', state='CA')`, c’est-à-dire une partition US and CA.

Il est également recommandé de cocher la case Create partition if not exist qui apparaît afin de vous assurer de ne pas créer une partition en doublon.

Die on error
Cochez cette case pour arrêter le Job lorsqu’une erreur survient.

Advanced settings

Tez lib
Choisissez comment accéder aux bibliothèques de Tez :

- **Auto install** : lors de l’exécution, le Job charge et déploie les bibliothèques de Tez fournies par le Studio dans le répertoire spécifié dans le champ Install folder in HDFS, par exemple, `/tmp/usr/tez`.
  
  Si vous avez configuré la propriété `tez.lib.uris` dans la table des propriétés, ce répertoire écrase la valeur de la propriété lors de l’exécution. Les autres propriétés configurées dans la table sont toujours prises en compte.

- **Use exist** : le Job accède aux bibliothèques de Tez déjà déployées dans le cluster Hadoop à utiliser.
  
  Vous devez saisir le chemin d’accès pointant vers ces bibliothèques dans le champ Lib path (folder or file).

- **Lib jar** : cette table s’affiche lorsque vous avez sélectionné Auto install dans la liste Tez lib et que vous utilisez une distribution personnalisée.
| Temporary path | Si vous ne souhaitez pas configurer le Jobtracker et le NameNode lorsque vous exécutez la requête `select * from your_table_name`, vous devez paramétrer un chemin d'accès temporaire. Par exemple, `/C:/select_all` sous Windows. |
| Hadoop properties | Le Studio Talend utilise une configuration par défaut pour son moteur, afin d'effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l'exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.  
* Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste Property type, vous passiez de Repository à Built-in.  
  Pour plus d'informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d'Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d'exemple, les liens vers certaines propriétés sont listés ci-après :  
| Hive properties | Le Studio Talend utilise la configuration par défaut pour son moteur afin d'effectuer des opération dans un base de données Hive. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez cette table avec la (les) propriété(s) à personnaliser. Ensuite, à l'exécution, la (les) propriété(s) personnalisée(s) écrasent celles par défaut. Pour plus d'informations concernant les propriétés dédiées à Hive, consultez [https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration](https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration) (en anglais).  
* Si vous devez utiliser Tez pour exécuter votre Job Hive, ajoutez `hive.execution.engine` à la colonne Properties et `Tez` à la valeur Value, en entourant ces chaînes de caractères de guillemets doubles. |
- Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

| **Mapred job map memory mb** et **Mapred job reduce memory mb** | Vous pouvez personnaliser les opérations map et reduce en cochant la case **Set memory**, pour configurer les allocations de mémoire pour ces opérations à effectuer par le système Hadoop.

Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations *map* et *reduce* dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux 1000, ce qui est normalement adapté pour l'exécution de ces opérations.

Les paramètres de mémoire à définir sont **Map (in Mb)**, **Reduce (in Mb)** et **ApplicationMaster (in Mb)**. Ces champs permettent d'allouer dynamiquement de la mémoire aux opérations map et reduce et à l'ApplicationMaster de YARN.

| **Path separator in server** | Laissez le champ **Path separator in server** tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d'autres termes, changez le séparateur si celui-ci n'est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.

| **tStatCatcher Statistics** | Cochez cette case afin de collecter les données de log au niveau des composants.

**Global Variables**

| **Global Variables** | **QUERY** : requête traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
### Utilisation

| Règle d’utilisation | Ce composant est utilisé en standalone et supporte l’écriture d’une grande variété de formats de données comme les formats RC, ORC et AVRO.

Si le *Studio Talend* utilisé pour vous connecter à une base de données Hive fonctionne sous Windows, vous devez créer manuellement un dossier appelé *tmp* à la racine du disque où le *Studio Talend* est installé.

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table *Dynamic settings* est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue *Basic settings*. Lorsqu’un paramètre dynamique est configuré, la liste *Component List* de la vue *Basic settings* devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez *Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte* à la page 2641 et *Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement* à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

| Prérequis | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le *Studio Talend* et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées `MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native`. Par exemple, pour Windows, la bibliothèque est `lib\MapRClient.dll` dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).
Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante :

```
no MapRClient in java.library.path.
```

- Configurez l’argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]** dans le menu **Window**. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

**Scénario associé**

Pour un scénario associé, consultez **Scénario : Créer une table Hive partitionnée** à la page 1680.
tHiveRow

Selon la nature de la requête et de la base de données, ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler).

Le tHiveRow exécute des requêtes Hive QL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Ce composant peut également effectuer des requêtes dans une case de données HBase, une fois la case Store by HBase cochée.

Propriétés du tHiveRow Standard

Ces propriétés sont utilisées pour configurer le tHiveRow s’exécutant dans le framework de Jobs Standard.

Le composant tHiveRow Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est toujours disponible.

Basic settings

La configuration de la connexion :

- Lorsque vous utilisez ce composant avec Google Dataproc :

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster identifier</td>
<td>Saisissez l’ID de votre cluster Dataproc à utiliser.</td>
</tr>
<tr>
<td>Region</td>
<td>Saisissez les régions dans lesquelles sont utilisées les ressources de calcul et dans lesquelles sont stockées et traitées les données. Si vous n’avez pas besoin de spécifier une région en particulier, laissez la valeur par défaut global. Pour plus d’informations relatives aux régions disponibles et aux zones de chaque groupe de région, consultez Regions and Zones (en anglais).</td>
</tr>
<tr>
<td>Google Storage staging bucket</td>
<td>Comme un Job Talend nécessite ses fichiers .jar dépendants pour être exécuté, spécifiez le répertoire Google Storage dans lequel ces fichiers .jar sont transférés afin que votre Job accède à ces fichiers lors de l’exécution. Le répertoire à saisir doit se terminer par une barre oblique (/). Si le répertoire n’existe pas, un répertoire est créé à la volée mais le bucket à utiliser doit déjà exister.</td>
</tr>
</tbody>
</table>
### Database
Renseignez ce champ avec le nom de la base de données.

### Provide Google Credentials in file
Lorsque vous lancez votre Job à partir d’une machine donnée sur laquelle Google Cloud SDK a été installé et vous a autorisé à utiliser vos identifiants de compte utilisateur pour accéder à Google Cloud Platform, ne cochez pas cette case. Dans cette situation, cette machine est souvent votre machine locale.


• Lorsque vous utilisez ce composant avec HDInsight :

#### WebHCat configuration

Dans le champ **Job result folder**, saisissez l’emplacement où vous souhaitez stocker les résultats d’exécution du Job dans Azure Storage.

#### HDInsight configuration
Saisissez les informations d’authentification du cluster HD Insight à utiliser.

#### Windows Azure Storage configuration

Dans le champ **Container**, saisissez le nom du conteneur à utiliser.

Dans le champ **Deployment Blob**, saisissez l’emplacement où vous souhaitez stocker le Job et ses bibliothèques dépendantes dans le compte Azure Storage.

### Database
Renseignez ce champ avec le nom de la base de données.

• Lorsque vous utilisez les autres distributions :

#### Connection mode
 Sélectionnez un mode de connexion dans la liste. Les options varient en fonction de la distribution que vous utilisez.

#### Hive server
Sélectionnez le serveur Hive sur lequel vous souhaitez que le Job utilisant ce composant exécute des requêtes dans Hive.
La liste **Hive server** est disponible uniquement lorsque la distribution Hadoop à utiliser, par exemple **HortonWorks Data Platform V1.2.0 (Bimota)** supporte HiveServer2. Vous pouvez sélectionner HiveServer2 (**Hive 2**), le serveur supportant mieux les connexions simultanées de différents clients que HiveServer (**Hive 1**).


<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Renseignez ce champ avec le nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Données d’authentification de l’utilisateur de la base de données.</td>
</tr>
<tr>
<td><strong>Use kerberos authentication</strong></td>
<td>Si vous accédez au Metastore de Hive avec une sécurité Kerberos, cochez cette case et saisissez ensuite les paramètres appropriés dans les champs qui s’affichent.</td>
</tr>
<tr>
<td></td>
<td>• Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans <a href="#">Connexion sécurisée à MapR</a> à la page 1745.</td>
</tr>
<tr>
<td></td>
<td>Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases <strong>Force MapR ticket authentication</strong> et <strong>Use Kerberos authentication</strong>. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.</td>
</tr>
<tr>
<td></td>
<td>Les valeurs des paramètres suivants peuvent être trouvées dans le fichier <code>hive-site.xml</code> du système Hive utilisé.</td>
</tr>
</tbody>
</table>
1. **Hive principal** utilise la valeur de `hive.metastore.kerberos.principal`. C'est le principal du service du Metastore de Hive.

2. **HiveServer2 local user principal** utilise la valeur de `hive.server2.authentication.kerberos.principal`.

3. **HiveServer2 local user keytab** utilise la valeur de `hive.server2.authentication.kerberos.keytab`.

4. **Metastore URL** utilise la valeur de `javax.jdo.option.ConnectionURL`. C'est la chaîne JDBC de connexion au Metastore de Hive.

5. **Driver class** utilise la valeur de `javax.jdo.option.ConnectionDriverName`. C'est le nom du pilote de la connexion JDBC.

6. **Username** utilise la valeur de `javax.jdo.option.ConnectionUserName`. Ce paramètre, ainsi que le paramètre **Password**, sont utilisés pour les informations de connexion de l'utilisateur au Metastore de Hive.

7. **Password** utilise la valeur de `javax.jdo.option.ConnectionPassword`. Cette case est disponible ou indisponible selon la distribution d'Hadoop à laquelle vous vous connectez.

---

**Use a keytab to authenticate**

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l'aide d'un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est `user1` et le principal à utiliser est `guest`. Dans cette situation, assurez-vous que `user1` a les droits de lecture pour le fichier keytab à utiliser.

---

**Use SSL encryption**

Cochez cette case pour activer la connexion chiffrée SSL ou TLS.

Les champs qui s'affichent ensuite fournissent les informations d'authentification :

- Dans le champ **Trust store path**, saisissez le chemin ou parcourez votre système jusqu'au fichier TrustStore à utiliser. Par défaut, les types TrustStore supportés sont JKS et PKCS 12.
- Pour saisir le mot de passe, cliquez sur le bouton `[...]` à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe.
entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Cette fonctionnalité n’est disponible que pour HiveServer2 en mode **Standalone** pour les distributions suivantes :
- Hortonworks Data Platform 2.0 +
- Cloudera CDH4 +
- Pivotal HD 2.0 +
- Amazon EMR 4.0.0 +

<table>
<thead>
<tr>
<th>Set Resource Manager</th>
</tr>
</thead>
</table>
| Cochez cette case et, dans le champ qui s’affiche, saisissez l’emplacement du ResourceManager de votre distribution. Par exemple `tal-qall4.tale nd.1an:8050`. Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d’un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du de l’exécution) :
| 1. Cochez la case **Set resourcemanager scheduler address** et saisissez l’adresse de l’ordonnanceur (Scheduler) dans le champ qui apparaît. |
| 2. Cochez la case **Set jobhistory address** et saisissez l’emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory. |
| 3. Cochez la case **Set staging directory** et saisissez le chemin d’accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l’exécution de programmes. Ce répertoire se trouve sous la propriété `yarn.app.mapreduce.am.staging-dir` dans les fichiers de configuration, notamment les fichiers `yarn-site.xml` et `mapred-site.xml` de votre distribution. |
| 4. Allouez des volumes de mémoire aux calculs **Map** et **Reduce** et au service **ApplicationMaster** de YARN en cochant la case **Set memory** dans la vue **Advanced settings**. |
| 5. Cochez la case **Set Hadoop user** et saisissez le nom de l’utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d’écriture, ce champ vous permet d’exécuter le Job directement avec l’utilisateur ayant les droits d’accès appropriés au fichier ou répertoire à traiter. |
| 6. Cochez la case **Use datanode hostname** pour permettre au Job d’accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété `dfs.client.use.datanode.hostname` à **true**. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case. |

**Set NameNode URI**

Cochez cette case et, dans le champ qui s'affiche, saisissez l'URI du NameNode Hadoop, le nœud maître d'un système Hadoop. Par exemple, si vous avez choisi une machine nommée `masternode` en tant que NameNode, l'emplacement est `hdfs://masternode:portnumber`. Si vous utilisez WebHDFS, l'emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL, le schéma d'URI doit être `swebhdfs` et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.


Les autres propriétés :

| Property type | Peut être **Built-In** ou **Repository**.  
**Built-In** : propriétés utilisées ponctuellement.  
**Repository** : sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |
|---------------|--------------------------------------------------------------------------------------------------------------------------|

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.  
**Remarque** :  
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :  
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.  
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.  
Pour plus d'informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**. |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------|

| Distribution | Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des |
composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center ([https://help.talend.com](https://help.talend.com)).

- L’option **Custom** vous permet de vous connecter à un cluster différente des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste [Hadoop configuration](en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend, Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’im
porter les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

### Hive version

Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.

### Execution engine

Cochez cette case et, dans la liste déroulante, sélectionnez le framework à utiliser pour exécuter le Job.

Cette liste est disponible lorsque vous utilisez le mode Embedded pour la connexion et distribution Hive avec laquelle vous travaillez, parmi les suivantes :

- Hortonworks : V2.1 et V2.2.
- MapR : V4.0.1.
- **Custom** : cette option vous permet de vous connecter à une distribution supportant mais non officiellement supportée par Talend.

Avant d’utiliser Tez, vérifiez que votre cluster Hadoop supporte Tez. Vous devez configurer l’accès aux bibliothèques Tez correspondantes via la vue Advanced settings de ce composant.

Pour plus d’informations concernant Hive avec Tez, consultez la documentation Apache à l’adresse [https://cwiki.apache.org/confluence/display/Hive/Hive+on+Tez](https://cwiki.apache.org/confluence/display/Hive/Hive+on+Tez) (en anglais). Des exemples vous sont présentés afin d’expliquer comment Tez peut être utilisé pour optimiser les performances par rapport à MapReduce.

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la
<table>
<thead>
<tr>
<th>Tableau d'informations</th>
<th>Tableau d'informations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>métadonnée du schéma dans la fenêtre [Repository Content].</strong></td>
<td><strong>Built-in</strong> : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Table Name</strong></td>
<td>Nom de la table à traiter.</td>
</tr>
<tr>
<td><strong>Query type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.</td>
<td><strong>Repository</strong> : Sélectionnez la requête appropriée dans le Repository. Le champ <strong>Query</strong> est renseigné automatiquement.</td>
</tr>
<tr>
<td><strong>Guess Query</strong></td>
<td>Cliquez sur le bouton <strong>Guess Query</strong> pour générer la requête correspondant au schéma de votre table dans le champ <strong>Query</strong>.</td>
</tr>
<tr>
<td><strong>This query uses Parquet objects</strong></td>
<td>Lorsqu'elle est disponible, cochez cette case pour indiquer que la table à gérer utilise le format Parquet et permet au composant d’appeler le fichier Jar requis. Notez que, lorsque le format de fichier à utiliser est <strong>PARQUET</strong>, il est possible qu’il vous soit demandé de trouver le fichier Jar Parquet spécifique et l’installer dans le studio.</td>
</tr>
<tr>
<td></td>
<td>• Lorsque le mode de connexion à Hive est <strong>Embedded</strong>, le Job est exécuté sur votre machine locale et appelle ce Jar installé dans le studio.</td>
</tr>
<tr>
<td></td>
<td>• Lorsque le mode de connexion à Hive est <strong>Standalone</strong>, le Job est exécuté sur le serveur hébergeant Hive et ce fichier Jar est envoyé au système HDFS du cluster auquel vous vous connectez. Assurez-vous d’avoir correctement défini l’URI du NameNode dans le champ correspondant de la vue <strong>Basic settings</strong>.</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma. Pour plus d’informations concernant le langage de requêtes Hive, consultez <a href="https://cwiki.apache.org/confluence/display/Hive/LanguageManual">https://cwiki.apache.org/confluence/display/Hive/LanguageManual</a> (en anglais).</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
</tbody>
</table>

Hadoop fournit différents formats de compression permettant de réduire l’espace nécessaire au stockage des fichiers et d’accélérer le transfert de données. Lorsque vous lisez un fichier comprimé, le Studio Talend doit le décompresser avant de pouvoir en alimenter le flux d’entrée.

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

**Store by HBase**

Cochez cette case afin d’afficher les paramètres à configurer pour permettre aux composants Hive d’accéder aux tables HBase :

- Une fois l’accès configuré, vous pourrez utiliser, dans un **tHiveRow** et un **tHiveInput**, les instructions Hive QL permettant de lire et d’écrire des données dans HBase.
- Si vous utilisez l’authentification Kerberos, vous devez définir les principaux relatifs à HBase dans les champs correspondants qui sont affichés.

Pour plus d’informations à propos de cet accès concernant Hive et HBase, consultez la documentation de Apache Hive concernant l’intégration Hive/HBase.

**Zookeeper quorum**

Saisissez le nom ou l’URL du service Zookeeper utilisé pour coordonner les transactions entre votre Studio et votre base de données. Notez que, lorsque vous configurez Zookeeper, vous pouvez avoir besoin de configurer explicitement la propriété **zookeeper.znode.parent** pour définir le chemin vers le nœud znode racine contenant tous les znodes créés et utilisés par votre base de données. Cochez la case la case **Set Zookeeper znode parent** afin de définir cette propriété.

**Zookeeper client port**

Saisissez le numéro du port d’écoute client du service Zookeeper que vous utilisez.

**Define the jars to register for HBase**

Cochez cette case pour afficher la table **Register jar for HBase**, dans laquelle vous pouvez enregistrer tout fichier Jar manquant, requis pour HBase, par exemple, Hive Storage Handler, enregistré par défaut avec votre installation Hive.

**Register jar for HBase**

Cliquez sur le bouton [+ ] pour ajouter des lignes à la table, puis, dans la colonne **Jar name**, sélectionnez le(s)
**fichier(s) Jar à enregistrer. Dans la colonne **Jar path**, saisissez le chemin d’accès à ce(s) Jar(s).**

**Advanced settings**

| Tez lib | Choisissez comment accéder aux bibliothèques de Tez :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>Auto install</strong> : lors de l’exécution, le Job charge et déploie les bibliothèques de Tez fournies par le Studio dans le répertoire spécifié dans le champ  Install folder in HDFS, par exemple, /tmp/usr/tez. Si vous avez configuré la propriété tez.lib.uris dans la table des propriétés, ce répertoire écrase la valeur de la propriété lors de l’exécution. Les autres propriétés configurées dans la table sont toujours prises en compte.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Use exist</strong> : le Job accède aux bibliothèques de Tez déjà déployées dans le cluster Hadoop à utiliser. Vous devez saisir le chemin d’accès pointant vers ces bibliothèques dans le champ Lib path (folder or file).</td>
<td></td>
</tr>
<tr>
<td>• <strong>Lib jar</strong> : cette table s’affiche lorsque vous avez sélectionné <strong>Auto install</strong> dans la liste <strong>Tez lib</strong> et que vous utilisez une distribution personnalisée (Custom). Dans cette table, vous devez ajouter les bibliothèques de Tez à charger.</td>
<td></td>
</tr>
</tbody>
</table>

| Temporary path | Si vous ne souhaitez pas configurer le Jobtracker et le NameNode lorsque vous exécutez la requête select * from your_table_name, vous devez paramétrer un chemin d’accès temporaire. Par exemple, /C:/select_all sous Windows. |

| Propagate QUERY’s recordset | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column. |

*Remarque :* Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type Object. Ce composant est généralement suivi du tParseRecordSet.

| Hadoop properties | Le Studio Talend utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut. |

*Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce...*
que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :


### Hive properties

Le **Studio Talend** utilise la configuration par défaut pour son moteur afin d’effectuer des opérations dans un base de données Hive. Si you devez utiliser une configuration personnalisée dans une situation spécifique, renseignez cette table avec la (les) propriété(s) à personnaliser. Ensuite, à l’exécution, la (les) propriété(s) personnalisée(s) écrasent celles par défaut. Pour plus d’informations concernant les propriétés dédiées à Hive, consultez [https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration](https://cwiki.apache.org/confluence/display/Hive/AdminManual+Configuration) (en anglais).

- Si vous devez utiliser Tez pour exécuter votre Job Hive, ajoutez `hive.execution.engine` à la colonne **Properties** et `Tez` à la valeur **Value**, en entourant ces chaînes de caractères de guillemets doubles.
- Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

### Mapred job map memory mb et Mapred job reduce memory mb

Vous pouvez personnaliser les opérations map et reduce en cochant la case **Set memory**, pour configurer les allocations de mémoire pour ces opérations à effectuer par le système Hadoop.

Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations map et reduce dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux 1000, ce qui est normalement adapté pour l’exécution de ces opérations.
Path separator in server

Laissez le champ **Path separator in server** tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable **PATH**. En d’autres termes, changez le séparateur si celui-ci n’est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Variables globales

| Global Variables | QUERY : requête traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.  
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes Hive QL.  
Le tHiveRow peut capturer les valeurs Application_ID et les écrire dans les logs du Job une fois que vous avez activé Log4j et configuré son niveau de sortie à Info pour votre Job utilisant un tHiveRow.  
• Pour plus d’informations concernant la définition du niveau de sortie Log4j au niveau d’un Job individuel, recherchez Personnaliser le niveau de sortie de Log4j à l’exécution, sur Talend Help Center (https://help.talend.com).  
• Pour plus d’informations concernant la configuration de Log4j au niveau du Studio pour appliquer la configuration à tous les Jobs, recherchez Configurer Log4j, sur Talend Help Center (https://help.talend.com).  
Si le **Studio Talend** utilisé pour vous connecter à une base de données Hive fonctionne sous Windows, vous |
devez créer manuellement un dossier appelé tmp à la racine du disque où le Studio Talend est installé.

**HBase Configuration**

<table>
<thead>
<tr>
<th>Remarque :</th>
<th>Store by HBase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disponible uniquement lorsque la case <em>Use an existing connection</em> est décochée.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zookeeper quorum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zookeeper client port</td>
</tr>
<tr>
<td>Define the jars to register for HBase</td>
</tr>
<tr>
<td>Register jar for HBase</td>
</tr>
</tbody>
</table>

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table *Dynamic settings* est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue *Basic settings*. Lorsqu’un paramètre dynamique est configuré, la liste *Component List* de la vue *Basic settings* devient inutilisable.


**Prérequis**

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les bibliothèques du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client.
Connexion sécurisée à MapR

Lorsque vous créez un Job, paramétrez la configuration d’authentification dans le composant que vous utilisez selon la sécurité de votre cluster MapR.

MapR supporte les deux méthodes suivantes d’authentification des utilisateurs et générant un ticket de sécurité MapR pour un utilisateur : une paire identifiant/mot de passe et Kerberos.

Pour plus d’informations concernant les mécanismes de sécurité de MapR, consultez MapR security architecture (en anglais).

Pour un scénario expliquant comment sécuriser un cluster MapR, consultez Getting started with MapR security (en anglais).

Les différents scénarios de sécurité auxquels vous pouvez être confrontés avec votre cluster MapR sont les suivants :

- Lorsque votre cluster MapR cluster sécurisé uniquement via Kerberos, vous devez seulement paramétrer la configuration de Hadoop Kerberos pour votre Job dans le Studio.

- Lorsque votre cluster MapR est sécurisé via Kerberos et le mécanisme de ticket de sécurité de MapR, vous devez paramétrer en conséquence leur configuration pour le Job dans le Studio.

Vous pouvez également trouver un exemple de configuration de Kerberbos pour un Job Talend Job dans Talend Help Center (https://help.talend.com).

Même si cet exemple utilise Cloudera à des fins de démonstration, les opérations décrites sont génériques et s’appliquent également à MapR.
Configurer l’authentification par ticket à MapR

Avant de commencer

- Votre distribution MapR utilise la version 4.0.1 ou une version supérieure et vous devez l’avoir sélectionnée comme cluster auquel vous connecter dans le composant à configurer.

- Le cluster MapR a été bien installé et est en cours d’exécution.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

  Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Cette section explique uniquement les paramètres d’authentification à utiliser pour vous connecter à MapR. Vous devez définir les autres paramètres requis par votre Job en plus de ceux d’authentification.

  Pour plus d’informations, consultez la documentation relative aux composants que vous utilisez.

Pourquoi et quand exécuter cette tâche

Dans un Job standard, vous devez paramétrer cette configuration dans l’onglet Basic settings d’un composant relatif à Hadoop à utiliser dans votre Job.

Dans cet onglet, procédez comme suit :

Procédure

1. Cochez la case Force MapR ticket authentication pour afficher les paramètres associés à définir.

2. Dans le champ Username, saisissez le nom de l’utilisateur à authentifier et, dans le champ Password, spécifiez le mot de passe utilisé par l’utilisateur.

   Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

   Un ticket de sécurité MapR est généré pour cet utilisateur par MapR et stocké sur la machine sur laquelle le Job que vous configurez est exécuté.

3. Si le champ Group est disponible dans cet onglet, vous devez saisir le nom du groupe auquel l’utilisateur à authentifier appartient.

4. Dans le champ Cluster name, saisissez le nom du cluster MapR auquel vous souhaitez que l’utilisateur se connecte.

   Le nom du cluster se trouve dans le fichier mapr-clusters.conf situé dans le dossier /opt/mapr/conf du cluster.

5. Dans le champ Ticket duration, saisissez la durée du temps (en secondes) durant laquelle le ticket est valide.
Utiliser une configuration de sécurité MapR personnalisée (facultatif)

Si la configuration par défaut de la sécurité de votre cluster MapR a été modifiée, vous devez configurer le Job à exécuter pour prendre en compte cette configuration personnalisée de sécurité.

MapR spécifie sa configuration de sécurité dans le fichier de configuration dans le fichier `mapr.login.conf` situé dans le dossier `/opt/mapr/conf` du cluster. Pour plus d'informations concernant ce fichier de configuration et les services Java qu'il utilise, consultez `mapr.login.conf` et `JAAS` (pages en anglais).

Pourquoi et quand exécuter cette tâche

Pour configurer votre Job, vous devez définir les paramètres associés dans les onglets Basic settings et Advanced settings de la vue Component du composant que vous souhaitez que votre Job utilise pour se connecter à MapR.

Procédez comme suit pour configurer :

**Procédure**

1. Vérifiez ce qui a été modifié dans le fichier `mapr.login.conf`.
   Vous devriez pouvoir obtenir les informations relatives de l'administrateur ou du développeur de votre cluster MapR.

2. Si l'emplacement de vos fichiers de configuration MapR a été modifié dans le cluster, c'est-à-dire si le répertoire Home MapR a été modifié cochez la case Set the MapR Home directory et saisissez le nouveau répertoire Home. Sinon, laissez la case décochée, pour utiliser le répertoire Home par défaut.

3. Si le module d’identification à utiliser dans le fichier `mapr.login.conf` a été modifié, cochez la case Specify the Hadoop login configuration et saisissez le module à appeler depuis le fichier `mapr.login.conf`. Sinon, laissez décochée cette case pour utiliser le module d’authentification par défaut.
   Par exemple, saisissez `kerberos` pour appeler le module `hadoop_kerberos` ou `hybrid` pour appeler le module `hadoop_hybrid`.

**Scénarios associés**

Pour un scénario associé, consultez :

- Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
- Procédure du composant tDBSQLRow.
- Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du composant tMysqlRow.

Lors de la configuration du composant devant se connecter à une distribution Hadoop, gardez à l'esprit que des paramètres sont requis par Hadoop, tels que le NameNode et le Jobtracker.
tHSQLDblnput

Ce composant exécute une requête de base de données dans un ordre strictement défini devant correspondre à la définition du schéma. Il passe ensuite la liste des champs au composant suivant à l’aide d’un lien Main row.

Le tHSQLDblnput lit les données d’une base de données et extrait les champs selon la requête exécutée.

Propriétés du tHSQLDblnput Standard

Ces propriétés sont utilisées pour configurer le tHSQLDblnput s’exécutant dans le framework de Jobs Standard.

Le composant tHSQLDblnput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

| Running Mode | Sélectionnez dans la liste le type de serveur correspondant aux paramètres de votre base de données, parmi les quatre proposés : HSQLDb Server, HSQLDb WebServer, HSQLDb In Process Persistent, HSQLDb In Memory. |

<table>
<thead>
<tr>
<th>Use TLS/SSL sockets</th>
<th>Cochez cette case pour autoriser, si besoin, le mode sécurisé.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database Alias</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
</tbody>
</table>
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

### DB path
Spécifiez le chemin d’accès à la base de données à laquelle vous souhaitez vous connecter. Ce champ n’est disponible qu’en mode d’exécution **HSQLDb In Process Persistent**.

**Remarque :**
Par défaut, si la base de données que vous spécifiez dans ce champ n’existe pas, elle sera créée automatiquement. Si vous souhaitez changer la configuration par défaut, modifiez le paramètre de connexion défini dans le champ **Additional JDBC parameters** dans la vue **Advanced settings**.

### Db name
Saisissez le nom de la base de données à laquelle vous souhaitez vous connecter. Ce champ n’est disponible qu’avec les modes d’exécution **HSQLDb In Process Persistent** et **HSQLDb In Memory**.

### Schema et Edit Schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

**Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</th>
</tr>
</thead>
</table>
| Query type | Peut être Built-in ou Repository.  
**Built-in** : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.  
**Repository** : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement. |
| Guess Query | Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**. |
| Guess schema | Cliquez sur le bouton pour récupérer le schéma de la table. |
| Query | Saisissez votre requête en respectant l’ordre des champs défini dans le schéma. |

**Advanced settings**

<table>
<thead>
<tr>
<th>Additional JDBC parameters</th>
<th>Spécifiez des informations supplémentaires de connexion à la base de données créée. Lorsque le mode d’exécution est <strong>HSQLDb In Process Persistent</strong>, vous pouvez configurer le paramètre de connexion <strong>ifexists=true</strong> afin de permettre la connexion à une base de données existante uniquement et éviter de créer une nouvelle base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim all the String/Char columns</td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td>Trim column</td>
<td>Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| Global Variables | **NB_LINE** : Indique le nombre de lignes traitées. C’est un variable After et elle retourne un entier.  
**QUERY** : Indique la requête à traiter. C’est une variable Flow et elle retourne une chaîne de caractères.  
Pour plus d’informations sur les variables, voir [Studio Talend User Guide](https://www.talend.com). |
|------------------|-------------------------------------------------------------------------------------------------
|                  | Remarque :  
A Flow variable means it functions during the execution of a component while an After variable means it functions after the execution of a component. |
## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requêtes SQL sur une base de données HSQLDb.</th>
</tr>
</thead>
</table>

### Connections

<table>
<thead>
<tr>
<th>Liens de sortie (de ce composant à un autre) :</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row</strong> : Main, Iterate.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Liens d’entrée (d’un autre composant à celui-ci) :</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row</strong> : Iterate.</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant les connexions, consultez la section relatives aux types de connexions, dans le *Guide utilisateur du Studio Talend*.

### Limitation


## Scénarios associés

Pour des scénarios associés, consultez :
**tHSQLDbOutput**

Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Le tHSQLDbOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

**Propriétés du tHSQLDbOutput Standard**

Ces propriétés sont utilisées pour configurer le tHSQLDbOutput s’exécutant dans le framework de Jobs Standard.

Le composant tHSQLDbOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Running Mode**

 Sélectionnez dans la liste le type de serveur correspondant aux paramètres de votre base de données, parmi les quatre proposés :

- HSQLDb Server, HSQLDb WebServer, HSQLDb In Process Persistent, HSQLDb In Memory.

**Use TLS/SSL sockets**

Cochez cette case pour autoriser, si besoin, le mode sécurisé.

**Host**

Adresse IP du serveur de base de données.

**Port**

Numéro du port d’écoute du serveur.

**Database**

Nom de la base de données.

**Username et Password**

Informations d’authentification de l’utilisateur de base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue
qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

### DB path
Spécifiez le chemin d’accès à la base de données à laquelle vous souhaitez vous connecter. Ce champ n’est disponible qu’en mode d’exécution **HSQLDb In Process Persistent**.

**Remarque :**
Par défaut, si la base de données que vous spécifiez dans ce champ n’existe pas, elle sera créée automatiquement. Si vous souhaitez changer la configuration par défaut, modifiez le paramètre de connexion défini dans le champ **Additional JDBC parameters** dans la vue **Advanced settings**.

### Db name
Saisissez le nom de la base de données à laquelle vous souhaitez vous connecter. Ce champ n’est disponible qu’avec les modes d’exécution **HSQLDb In Process Persistent** et **HSQLDb In Memory**.

### Table
Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.

### Action on table
Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :

- **None** : n’effectuer aucune opération de table.
- **Drop and create the table** : supprimer la table puis en créer une nouvelle.
- **Create a table** : créer une table qui n’existe pas encore.
- **Create table if doesn’t exist** : créer la table si nécessaire.
- **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.
- **Clear a table** : supprimer le contenu de la table.

### Action on data
Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

- **Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
- **Update** : Mettre à jour les entrées existantes.
- **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
- **Delete** : Supprimer les entrées correspondantes au flux d’entrée.

**Avertissement :**
Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle
baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

<table>
<thead>
<tr>
<th><strong>Schema et Edit schema</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être <strong>Built-in</strong> ou distant dans le <strong>Repository</strong>. Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td><strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td><strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
<tr>
<td><strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <strong>No</strong> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre <strong>[Repository Content]</strong>.</td>
</tr>
</tbody>
</table>

| **Built-In** | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**. |

| **Repository** | Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**. Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets. Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com). |

| **Die on error** | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le Job. |
trajetement avec les lignes sans erreur, et ignorer les
lignes en erreur. Vous pouvez récupérer les lignes en
erreur, si vous le souhaitez. Pour cela, utilisez un lien
Row > Rejects.

### Advanced settings

| **Additional JDBC parameters** | Spécifiez des informations supplémentaires de
connexion à la base de données créée. Lorsque le
mode d’exécution est HSQLDb In Process Persistent,
 cette propriété supplémentaire est définie comme
ifexists=true par défaut, ce qui signifie que la base
de données sera automatiquement créée lorsque c’est
nécessaire.

**Remarque :**
Vous pouvez appuyer sur Ctrl+Espace afin
d’accéder à une liste de variables globales
pré définies.

| **Commit every** | Nombre de lignes à inclure dans le lot avant de
commencer l’écriture dans la base. Cette option garantit
la qualité de la transaction (cependant pas de rollback)
et surtout une meilleure performance d’exécution.

| **Additional Columns** | Cette option n’est pas disponible si vous venez de créer
la table de données (que vous l’ayez préalablement
supprimée ou non). Cette option vous permet d’effectuer
des actions sur les colonnes, à l’exclusion des actions
d’insertion, de mise à jour, de suppression ou qui
nécessitent un prétraitement particulier.

| **Name** | Saisissez le nom de la colonne à modifier ou à
insérer.

| **SQL expression** | Saisissez la déclaration SQL à exécuter
pour modifier ou insérer les données dans les colonnes
correspondantes.

| **Position** | Sélectionnez Before, Replace ou After, en
fonction de l’action à effectuer sur la colonne de
référence.

| **Reference column** | Saisissez une colonne de référence
que le composant tHSQLDbOutput peut utiliser pour
situer ou remplacer la nouvelle colonne ou celle à
modifier.

**Use field options** | Cochez cette case pour personnaliser une requête,
surtout lorsqu’il y a plusieurs actions sur les données.

**Enable debug mode** | Cochez cette case pour afficher chaque étape du
processus d’écriture dans la base de données.

**tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au
niveau du composant.
**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong></td>
<td>nombre de lignes mises à jour. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong></td>
<td>nombre de lignes insérées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong></td>
<td>nombre de lignes supprimées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_REJECTED</strong></td>
<td>nombre de lignes rejetées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>QUERY</strong></td>
<td>requête traitée. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données HSQLDb. Il permet aussi de créer un flux de rejet avec un lien <em>Row &gt; Reject</em> filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien <em>Reject</em> à la page 2675 du composant <em>tMysqlOutput</em>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connections</th>
<th>Liens de sortie (de ce composant à un autre) :</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Row</strong> : Main, Iterate.</td>
</tr>
<tr>
<td></td>
<td>Liens d’entrée (d’un autre composant à celui-ci) :</td>
</tr>
<tr>
<td></td>
<td><strong>Row</strong> : Main.</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant les connexions, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*.
**Limitation**

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

---

**Scénarios associés**

Pour un scénario associé, consultez :

- **Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput** à la page 2667 du composant **tMysqlOutput**.
tHSQLDbRow

Selon la nature de la requête et de la base de données, tHSQLDbRow agit sur la structure même de la base de données ou sur les données (mais sans les manipuler).

Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Le tHSQLDbRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tHSQLDbRow Standard

Ces propriétés sont utilisées pour configurer le tHSQLDbRow s’exécutant dans le framework de Jobs Standard.

Le composant tHSQLDbRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Running Mode**

 Sélectionnez dans la liste le type de serveur correspondant aux paramètres de votre base de données, parmi les quatre proposés :

- HSQLDb Server, HSQLDb WebServer, HSQLDb In Process Persistent, HSQLDb In Memory.

**Use TLS/SSL sockets**

 Cochez cette case pour autoriser, si besoin, le mode sécurisé.

**Host**

 Adresse IP du serveur de base de données.

**Port**

 Numéro du port d’écoute du serveur.

**Database Alias**

 Nom de la base de données.

**Username et Password**

 Informations d’authentification de l’utilisateur de base de données.

 Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.
| **DB path** | Spécifiez le chemin d'accès à la base de données à laquelle vous souhaitez vous connecter. Ce champ n'est disponible qu'en mode d'exécution HSQLDb In Process Persistent. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. |
| **Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. | |
| Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles : | |
| • **View schema** : sélectionnez cette option afin de voir le schéma. | |
| • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales. | |
| • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. | |
| **Query type** | Peut être Built-in ou Repository. |
| **Built-in** : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder. | |
| **Repository** : Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement. | |
| **Guess Query** | Cliquez sur le bouton Guess Query pour générer la requête correspondant au schéma de votre table dans le champ Query. |
| **Query** | Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma. |
| **Die on error** | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects. |

### Advanced settings

| **Additional JDBC parameters** | Spécifiez des informations supplémentaires de connexion à la base de données créée. Lorsque le mode d’exécution est HSQLDb In Process Persistent, vous pouvez configurer le paramètre de connexion ifexists=true afin de permettre la connexion à une base de données existante uniquement et éviter de créer une nouvelle base de données. |
| **Propagate QUERY’s recordset** | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column. |
**Parameter Index** : Saisissez la position du paramètre dans l’instruction SQL.  
**Parameter Type** : Saisissez le type du paramètre.  
**Parameter Value** : Saisissez la valeur du paramètre.  
**Remarque** : Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance. |
| **Commit every** | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| **Global Variables** | **QUERY** : indique le nombre de requêtes traitées. Cette variable est une variable Flow et retourne une chaîne de caractères. |
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du *Studio Talend*.

**Remarque :**
Une variable **Flow** signifie qu’elle fonctionne durant l’exécution d’un composant. Une variable **After** signifie qu’elle fonctionne après l’exécution d’un composant.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
</table>
| **Connections**     | Liens de sortie (de ce composant à un autre) :
Row : Main, Reject, Iterate.  
|                     | Liens d’entrée (d’un autre composant à celui-ci) :
Row : Main, Iterate.  
|                     | Pour plus d’informations concernant les connexions, consultez la section relative aux types de connexions, dans le *Guide utilisateur du Studio Talend*. |

| **Limitation** | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center ([https://help.talend.com](https://help.talend.com)). |

### Scénarios associés

Pour des scénarios associés, consultez :

- **Procédure.**
- **Scénario : Supprimer et re-générer un index de table MySQL à la page 2700.**
tHttpRequest

Envoie des requêtes HTTP au serveur et d'écrire les réponses localement.

*tHttpRequest* envoie une requête HTTP au serveur et obtient des réponses HTTP du serveur.

**Propriétés du tHttpRequest Standard**

Ces propriétés sont utilisées pour configurer le tHttpRequest s'exécutant dans le framework de Jobs Standard.

Le composant tHttpRequest Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|---|---|
| • View schema : sélectionnez cette option afin de voir le schéma.  
| • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
| • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| Sync columns | Cliquez sur ce bouton pour récupérer le schéma du composant précédent. |
| URI | Saisissez l’URI (Identifiant uniforme de ressource) à partir duquel seront identifiées les ressources sur le serveur. Un URI est similaire à un URL mais il est plus générique. |
| **Method** | Sélectionnez dans la liste la méthode pour définir l’action à effectuer :

**Post** : Choisissez cette méthode pour envoyer des données côté serveur (des données de formulaires HTML par exemple).

**Get** : Choisissez cette méthode pour récupérer des données côté serveur. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Post parameters from file</strong></td>
<td>Parcourez votre système ou entrez le chemin vers le fichier utilisé pour passer des paramètres (corps de la requête) à la méthode POST.</td>
</tr>
<tr>
<td><strong>Write response content to file</strong></td>
<td>Cochez cette case afin de sauvegarder la réponse HTTP dans un fichier local. Vous pouvez soit renseigner le chemin d’accès au fichier d’entrée, soit cliquer sur [...] pour choisir le chemin d’accès dans votre système de fichiers.</td>
</tr>
<tr>
<td><strong>Create directory if not exists</strong></td>
<td>Cochez cette case afin de créer le répertoire défini dans le champ <strong>Write response content to file</strong> s’il n’existe pas. Cette case est décochée par défaut et apparaît lorsque la case <strong>Write response content to file</strong> est cochée.</td>
</tr>
</tbody>
</table>
| **Headers** | Renseignez la (les) paire(s) nom-valeur des en-têtes HTTP pour définir les paramètres de l’opération HTTP requise :

**Key** : Saisissez le nom du champ d’en-tête de l’en-tête HTTP.

**Value** : Saisissez le contenu du champ d’en-tête de l’en-tête HTTP.

Pour plus d’informations concernant la définition des en-têtes HTTP, consultez (en anglais) :
en.wikipedia.org/wiki/List_of_HTTP_headers. |
| **Need authentication** | Cochez cette case afin de saisir un identifiant et un mot de passe dans les champs correspondants si une authentification est nécessaire :

**user** : Saisissez l’identifiant de connexion au serveur.

**password** : Saisissez le mot de passe de connexion au serveur.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Die on error** | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. |
### Advanced settings

| Set timeout | Cochez cette case pour spécifier les valeurs de suspension de la connexion et de la lecture dans les deux champs suivants :

- **Connect timeout(s)** : saisissez la valeur du délai avant suspension de la connexion, en secondes. Une exception est retournée si le délai avant suspension est expiré avant l’établissement de la connexion. La valeur 0 indique une délai infini. Par défaut, la valeur du délai avant suspension est de 30.

- **Read timeout(s)** : saisissez la valeur du délai avant suspension de la lecture, en secondes. Une exception est retournée si le délai avant suspension est expiré avant que les données soient disponibles à la lecture. Par défaut, la valeur du délai avant suspension de la lecture est de 0, ce qui indique un délai infini. |

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  
**CONNECTED** : indique si une connexion au serveur est établie. Cette variable est une variable *After* et retourne un booléen.  
**RESPONSE_CODE** : code de réponse retourné par le serveur distant HTTP. Cette variable est une variable *After* et retourne un nombre entier.  
Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend.* |

### Utilisation

| Règle d’utilisation | Ce composant peut être utilisé pour l’envoi de requêtes HTTP au serveur ainsi que pour l’enregistrement des réponses HTTP obtenues. Il peut être utilisé en standalone. |
Scénario 1 : Envoyer une requête HTTP au serveur et sauvegarder localement les réponses HTTP obtenues

Ce scénario décrivit un Job à deux composants qui utilise la méthode GET pour récupérer des informations du serveur et écrire les réponses dans un fichier local ainsi que dans la console.

Relier les composants

Procédure
   Pour plus d’informations concernant la création de Job, consultez le Guide utilisateur du Studio Talend.
2. A partir de la Palette, déposez les composants suivants dans l’espace de modélisation graphique : un composant tHttpRequest et un tLogRow.

![Diagramme de connexion tHttpRequest et tLogRow]

3. Connectez le composant tHttpRequest au composant tLogRow à l’aide d’une connexion de type Row > Main.

Configurer la requête GET

Procédure
1. Double-cliquez sur le composant tHttpRequest pour ouvrir l’onglet Basic settings de sa vue Component et paramétrer ses propriétés.

![Onglet Basic settings de tHttpRequest]

2. Saisissez "http://192.168.0.63:8081/testHttpRequest/build.xml" dans le champ URI. Notez que l’adresse URI fournie dans ce scénario est destinée essentiellement à la démonstration, ce n’est pas une adresse valide.

4. Cochez la case Write response content to file et renseignez le chemin d’accès dans le champ à droite. Pour ce scénario, saisissez D:/test.txt.

5. Cochez la case Need authentication. Pour ce scénario, saisissez “tomcat” dans les champs user et password.

**Exécuter le Job**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez à présent exécuter ce Job.

Le composant tLogRow est utilisé pour afficher le résultat de l’exécution de ce Job.

**Procédure**

1. Si vous souhaitez personnaliser la manière dont le tLogRow affiche les résultats, double-cliquez sur le composant pour ouvrir l’onglet Basic settings de sa vue Component et sélectionnez Table (print values in cells of a table) dans le champ Mode.

2. Appuyez sur F6 pour exécuter le Job.

**Résultats**

La réponse du serveur est alors sauvegardée et affichée.

```
[statistics] connecting to socket on port 4011
[statistics] connected

<table>
<thead>
<tr>
<th>tLogRow_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResponseContent</td>
</tr>
<tr>
<td>hello world!</td>
</tr>
</tbody>
</table>

[statistics] disconnected
```

**Scénario 2 : Envoyer une requête POST depuis un fichier local JSON**

Dans ce scénario, un Job de quatre composants est utilisé pour lire les paramètres depuis un fichier JSON et les envoyer à un site Web dans une requête POST.
Le fichier JSON utilisé se présente comme suit :

```json
{"echo":
  [
    {
      "data": "e=hello"
    }
  ]
}
```


Notez que le paramètre *e* est requis par `http://echo.itcuties.com/`.

**Relier les composants**

**Procédure**

   
   Pour plus d’informations concernant la création d’un Job, consultez le Guide utilisateur du Studio Talend.

2. Déposez un tFileInputJSON, un tFileOutputDelimited, un tHttpRequest et un tLogRow dans l’espace de modélisation graphique.

3. Reliez le tFileInputJSON au tHttpRequest à l’aide d’un lien Trigger > On Subjob Ok.

4. Reliez les autres composants entre eux à l’aide de liens Row > Main.

**Lire le fichier JSON**

**Procédure**

1. Double-cliquez sur le tFileInputJSON afin d’ouvrir sa vue Component.
2. Dans la liste Read By, sélectionnez JsonPath without loop.

3. Cliquez sur le bouton [...] à côté du champ Edit schema afin d’ouvrir l’éditeur de schéma.

4. Cliquez sur le bouton [+] afin d’ajouter une ligne et nommez-la, par exemple, data.

5. Cliquez sur OK afin de valider ces changements puis acceptez la propagation du schéma dans la boîte de dialogue qui s’affiche.

6. Dans le champ Filename, parcourrez votre système ou saisissez le chemin vers le fichier source JSON contenant les paramètres à envoyer.

7. Dans la table Mapping, la colonne data, définie précédemment dans le schéma du composant, est automatiquement ajoutée. Dans la colonne JSONPath query de cette table, saisissez le chemin JSON, entre guillemets, afin d’extraire les paramètres à envoyer. Dans ce scénario, le chemin est echo[0].data.

Écrire le paramètre dans un fichier plat

Procédure

1. Double-cliquez sur le tFileOutputDelimited afin d’ouvrir sa vue Component.
2. Dans le champ **File name**, parcourez votre système ou saisissez le chemin vers le fichier plat dans lequel vous souhaitez écrire les paramètres extraits. S'il n'existe pas, le fichier est créé à l'exécution. Dans cet exemple, le chemin est `C:/tmp/postParamsFile.txt`.

**Poster le paramètre**

**Procédure**

1. Double-cliquez sur le **tHttpRequest** afin d'ouvrir sa vue **Component**.


3. Dans la liste **Method**, sélectionnez **POST**.

4. Dans le champ **Post parameters from file**, parcourez votre système ou saisissez le chemin vers le fichier plat contenant les paramètres utilisés. Comme défini précédemment, dans le composant **tFileOutputDelimited**, ce chemin est `C:/tmp/postParamsFile.txt`.

**Exécuter le Job**

Appuyez sur **F6** pour exécuter ce Job.

Le **tLogRow** est utilisé pour afficher les résultats d'exécution du Job.

Une fois le Job exécuté, la vue **Run** s'ouvre automatiquement, vous permettant de vérifier le résultat de l'exécution.
Les réponses renvoyées par le site qui reçoit les paramètres s’affichent dans la console.
tImpalaClose

Ferme une connexion à une base de données Impala.
tImpalaClose ferme une connexion active pour une base de données Impala.

Propriétés du tImpalaClose Standard

Ces propriétés sont utilisées pour configurer le tImpalaClose s’exécutant dans le framework de Jobs Standard.
Le composant tImpalaClose Standard appartient à la famille Big Data.
Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Component list</th>
<th>Si plus d’une connexion est utilisée dans le Job, sélectionnez le tImpalaConnection dans la liste.</th>
</tr>
</thead>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Impala, notamment le tImpalaConnection, car le tImpalaConnection permet d’ouvrir une connexion pour la transaction en cours. |
### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend.

La liste suivante présente des informations d'exemple relatives à MapR.

- **Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine.** D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

  Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- **Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window.** Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.
Pour plus d'informations concernant l'installation d'une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Scénario associé

Aucun scénario n'est disponible pour la version Standard de ce composant.
**tImpalaConnection**

Etablit une connexion à Impala à réutiliser dans les autres composants Impala de votre Job.
tImpalaConnection ouvre une connexion à une base de données Impala.

**Propriétés du tImpalaConnection Standard**

Ces propriétés sont utilisées pour configurer le tImpalaConnection s’exécutant dans le framework de Jobs Standard.

Le composant tImpalaConnection Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Distribution**

 Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.

- Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).
- L’option Custom vous permet de vous connecter à un cluster différente des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,
2. Sélectionner **import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste [Hadoop configuration](en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend. Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’im importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez [Connexion à une distribution Hadoop personnalisée](en anglais) à la page 1677.

<table>
<thead>
<tr>
<th>Impala version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez dans ce champ le nom de la base de données.</td>
</tr>
<tr>
<td>Username</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td>Use kerberos authentication</td>
<td>Si vous accédez à un système Impala s’exécutant avec la sécurité Kerberos, cochez la case et saisissez le Principal Kerberos de ce système Impala.</td>
</tr>
</tbody>
</table>
Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**

Ce composant s’utilise avec d’autres composants Impala, notamment le `tImpalaClose`.

**Prérequis**

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le `Studio Talend`.

La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le `Studio Talend` et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client
MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

• Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

### Scénario associé

Ce composant s’utilise de la même manière que le composant tHiveConnection. Pour plus d’informations, consultez Scénario : Créer une table Hive partitionnée à la page 1680.
**tImpalaCreateTable**

Crée des tables Impala correspondant à différents formats de données Impala.

`tImpalaCreateTable` se connecte à une base de données à utiliser et crée une table Impala dédiée aux données du format spécifié.

### Propriétés du `tImpalaCreateTable Standard`

Ces propriétés sont utilisées pour configurer le `tImpalaCreateTable` s'exécutant dans le framework de Jobs Standard.

Le composant `tImpalaCreateTable Standard` appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

#### Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
</tbody>
</table>

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d'une connexion que vous avez déjà définie.

**Remarque :**

Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

**Distribution**

 Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des...
composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option **Custom** vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les Jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’im
porter les fichiers .jar correspondant à
la connexion créée entre la distribution
personnalisée et ce composant.

Pour un exemple étape par étape expliquant
court se connecter à une distribution
personnalisée et partager cette connexion, consultez
Connexion à une distribution Hadoop personnalisée
tà la page 1677.

| **Impala version** | Sélectionnez la version de la distribution Hadoop que
vous utilisez. Les options disponibles dépendent du
composant que vous utilisez. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Saisissez l’adresse IP du serveur de la base de données.</td>
</tr>
</tbody>
</table>
| **Port** | Saisissez le numéro du port d’écoute du serveur de la
base de données. |
| **Database** | Saisissez le nom de la base de données. |
| **Username et Password** | Saisissez les données d’authentification de l’utilisateur à
la base de données. |
| **Use kerberos authentication** | Si vous accédez à un système Impala s’exécutant avec la
sécurité Kerberos, cochez la case et saisissez le Principal
Kerberos de ce système Impala.

- Si ce cluster est un cluster MapR de version 5.0.0 ou
postérieure, vous pouvez paramétrer la configuration
de l’authentification par ticket MapR en plus ou
comme une alternative en suivant les explications
da Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère
un nouveau ticket de sécurité MapR pour le
nom d’utilisateur défini dans le Job dans chaque
exécution. Si vous devez réutiliser un ticket existant
provenant du même utilisateur, laissez décochées
les cases Force MapR ticket authentication et Use
Kerberos authentication. MapR devrait pouvoir
trouver automatiquement ce ticket à la volée.

Cette case est disponible ou indisponible selon la
distribution d’Hadoop à laquelle vous vous connectez. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le
nombre de champs (colonnes) qui sont traités et passés
au composant suivant. Lorsque vous créez un Job Spark,
évitez le mot réservé line lors du nommage des cham
ps.

Cliquez sur Edit schema pour modifier le schéma. Si
le schéma est en mode Repository, trois options sont
disponibles :

- **View schema** : sélectionnez cette option afin de voir
le schéma.

- **Change to built-in property** : sélectionnez cette
option pour passer le schéma en mode Built-In et
effectuer des modifications locales. |
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Saisissez le nom de la table à créer.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action on table</strong></td>
<td>Sélectionnez l'action à effectuer pour créer une table.</td>
</tr>
<tr>
<td><strong>Format</strong></td>
<td>Sélectionnez le format de données auquel la table à créer sera dédiée. Les formats de données disponibles varient selon la distribution Hadoop que vous utilisez. Notez que, lorsque le format de fichier à utiliser est <strong>PARQUET</strong>, il est possible qu'il vous soit demandé de trouver le fichier Jar Parquet spécifique et l'installer dans le studio.</td>
</tr>
<tr>
<td>• Lorsque le mode de connexion à Hive est <strong>Embedded</strong>, le Job est exécuté sur votre machine locale et appelle ce Jar installé dans le studio.</td>
<td></td>
</tr>
<tr>
<td>• Lorsque le mode de connexion à Hive est <strong>Standalone</strong>, le Job est exécuté sur le serveur hébergeant Hive et ce fichier Jar est envoyé au système HDFS du cluster auquel vous vous connectez. Assurez-vous d'avoir correctement défini l'URI du NameNode dans le champ correspondant de la vue <strong>Basic settings</strong>. Ce fichier Jar est téléchargeable depuis le site Web d'Apache. Vous pouvez trouver plus d'informations concernant l'installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
<td></td>
</tr>
<tr>
<td><strong>Set partitions</strong></td>
<td>Cochez cette case pour ajouter des colonnes de partition à la table à créer. Une fois cochée, vous devez définir le schéma des colonnes de partition à ajouter.</td>
</tr>
</tbody>
</table>
### Set file location

Si vous souhaitez créer une table Impala dans un répertoire différent de celui par défaut, cochez cette case et saisissez le chemin d'accès au répertoire HDFS à utiliser pour le contenu de la table.

Cela est utile lorsque vous devez créer une table Impala externe en cochant la case **Create an external table** dans l'onglet **Advanced settings**.

### Use S3 endpoint

La case **Use S3 endpoint** apparaît lorsque vous cochez la case **Set file location** pour créer une table Impala externe.

Après avoir coché la case **Use S3 endpoint**, vous devez renseigner les paramètres suivants dans les champs qui apparaissent :

- **S3 bucket** : saisissez le nom du bucket dans lequel vous souhaitez créer la table.
- **Access key** et **Secret key** : renseignez les informations d'authentification requises pour vous connecter au bucket Amazon S3 à utiliser.

Pour sauvegarder les paramètres, cliquez sur le bouton **[...]** à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Notez que le format du fichier S3 est S3N (S3 Native Filesystem).

Étant donné que la table Impala créée dans S3 est une table externe, la case **Use S3 endpoint** ne peut être utilisée qu'en cochant la case **Create an external table**.

### Advanced settings

#### Like table

Cochez cette case et saisissez le nom de la table Impala à copier. Cela vous permet de copier la définition d'une table existante sans copier ses données.

Pour plus d’informations concernant le paramètre Like, consultez la documentation Cloudera relative au langage de définition des données Impala.

#### Create an external table

Cochez cette case afin de faire de la table créée une table Hive externe. Ce type de table Hive laisse les données brutes où elles sont si les données sont dans HDFS.

Une table externe est habituellement le meilleur choix pour accéder aux données partagées existant dans un système de fichiers.

Pour plus d’informations concernant une table externe Impala, consultez la documentation Cloudera concernant Impala.

#### Table comment

Saisissez la description à utiliser pour la table à créer.
<table>
<thead>
<tr>
<th>As select</th>
<th>Cochez cette case et saisissez l'instruction <code>As select</code> pour créer une table Hive basée sur une instruction <code>Select</code>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table properties</td>
<td>Ajoutez une table de propriétés Impala personnalisée pour écraser les propriétés utilisées par défaut par le moteur Hadoop du studio.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| Global Variables | **ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable `Flow` fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

**Utilisation**

<table>
<thead>
<tr>
<th>Row format</th>
<th><strong>Set Delimited row format</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Die on error</td>
<td></td>
</tr>
</tbody>
</table>

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ `Code`, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez  *Scénario : Lire des données*. |
dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Usage</th>
<th>Ce composant s’utilise en standalone.</th>
</tr>
</thead>
</table>
| Prerequisites | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.  
  • Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.  
  • Configurez l’argument -Djava.library.path, par exemple, dans la zone Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR. Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez. |

**Scénario associé**

Ce composant s’utilise de la même manière que le tHiveCreateTable. Pour plus d’informations, consultez Scénario : Créer une table Hive partitionnée à la page 1680.
tImpalaInput

Exécute les requêtes SELECT pour extraire les données correspondantes et les envoyer au composant qui suit.

tImpalaInput est le composant dédié à la base de données Impala (système d’entrepôt de données Impala). Il fournit un outil de construction d’expression, le SQLBuilder afin de vous permettre d’écrire facilement vos instructions Impala SQL.

Propriétés du tImpalaInput Standard

Ces propriétés sont utilisées pour configurer le tImpalaInput s’exécutant dans le framework de Jobs Standard.

Le composant tImpalaInput Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.
Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez. Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.
Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Impala version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez dans ce champ le nom de la base de données.</td>
</tr>
<tr>
<td>Username</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td>Use kerberos authentication</td>
<td>Si vous accédez à un système Impala s’exécutant avec la sécurité Kerberos, cochez la case et saisissez le Principal Kerberos de ce système Impala.</td>
</tr>
<tr>
<td></td>
<td>• Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.</td>
</tr>
<tr>
<td></td>
<td>Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.</td>
</tr>
<tr>
<td></td>
<td>Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
</tbody>
</table>
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th>Built-in</th>
<th>Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</th>
</tr>
</thead>
</table>

**Table Name**
Nom de la table à traiter.

**Query type**
Peut être Built-in ou Repository.

- **Built-in** : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder

- **Repository** : Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.

**Guess Query**
Cliquez sur le bouton Guess Query pour générer la requête correspondant au schéma de votre table dans le champ Query.

**Guess schema**
Cliquez sur ce bouton pour récupérer le schéma de la table.

**Query**
Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.

### Advanced settings

<table>
<thead>
<tr>
<th>Trim all the String/Char columns</th>
<th>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaines de caractères.</th>
</tr>
</thead>
</table>
| Trim column | Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.  

**Remarque :**  
Décochez la case Trim all the String/Char columns pour activer l’option Trim column.
### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

|---|---|

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes Impala SQL.</th>
</tr>
</thead>
</table>

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les |
ImpalaInput

paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Scénario associé

Pour un scénario expliquant comment ce composant d’entrée est utilisé dans un Job, consultez Scénario : Ecrire des colonnes d’une base de données MySQL dans un fichier de sortie en utilisant tMysqlInput à la page 2635.
**tImpalaLoad**

Ecrit des données de différents formats dans une table Impala donnée ou pour exporter des données d’une table Impala vers un répertoire.

tImpalaLoad se connecte à une base de données Impala et copie ou déplace des données dans une table Impala existante ou dans un répertoire spécifié.

**Propriétés du tImpalaLoad Standard**

Ces propriétés sont utilisées pour configurer le tImpalaLoad s’exécutant dans le framework de Jobs Standard.

Le composant tImpalaLoad Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

**Distribution**

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que
vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Impala version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez dans ce champ le nom de la base de données.</td>
</tr>
<tr>
<td>Username</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td>Use kerberos authentication</td>
<td>Si vous accédez à un système Impala s’exécutant avec la sécurité Kerberos, cochez la case et saisissez le Principal Kerberos de ce système Impala.</td>
</tr>
<tr>
<td></td>
<td>• Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.</td>
</tr>
<tr>
<td></td>
<td>Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.</td>
</tr>
<tr>
<td></td>
<td>Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.</td>
</tr>
<tr>
<td>Load action</td>
<td>Sélectionnez l’action à effectuer pour écrire des données à l’emplacement spécifié.</td>
</tr>
<tr>
<td></td>
<td>• Lorsque vous sélectionnez LOAD, cela permet de déplacer ou copier des données d’un répertoire spécifié.</td>
</tr>
<tr>
<td></td>
<td>• Lorsque vous sélectionnez INSERT, cela permet de déplacer ou copier des données à partir de requêtes.</td>
</tr>
<tr>
<td>Target type</td>
<td>Cette liste déroulante apparaît uniquement lorsque vous avez sélectionné INSERT dans la liste Load action.</td>
</tr>
<tr>
<td></td>
<td>Sélectionnez dans cette liste le type d’emplacement dans lequel écrire des données.</td>
</tr>
</tbody>
</table>
Si vous sélectionnez **Table** comme destination, vous pouvez choisir d’écrire les données à la suite ou d’écraser le contenu dans la table spécifiée.

<table>
<thead>
<tr>
<th>Action</th>
<th>Sélectionnez <strong>OVERWRITE</strong> pour écraser les données existantes à l’emplacement de destination ou sélectionnez <strong>APPEND</strong> pour écrire les nouvelles données à la suite des anciennes.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table name</strong></td>
<td>Saisissez le nom de la table Impala dans laquelle écrire les données. Notez qu’avec l’action <strong>INSERT</strong>, ce champ est disponible uniquement lorsque vous avez sélectionné <strong>Table</strong> dans la liste <strong>Target type</strong>.</td>
</tr>
<tr>
<td><strong>File path</strong></td>
<td>Saisissez le répertoire duquel lire des données.</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Ce champ apparaît uniquement lorsque vous avez sélectionné <strong>INSERT</strong> dans la liste <strong>Load action</strong>.</td>
</tr>
<tr>
<td><strong>Set partitions</strong></td>
<td>Cochez cette case pour utiliser la clause Impala <strong>Partition</strong> pour charger ou insérer des données dans une table Impala. Vous devez saisir les clés de partition et leur valeur à utiliser dans le champ qui apparaît.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cochez cette case pour arrêter le Job lorsqu’une erreur survient.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>tStatCatcher Statistics</strong></th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th><strong>Global Variables</strong></th>
<th><strong>QUERY</strong> : requête traitée. Cette variable est une variable <strong>Flow</strong> et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en standalone.</th>
</tr>
</thead>
</table>
| **Dynamic settings** | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.


### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend.

La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les bibliothèques du client MapR correspondant à chaque OS peuvent être trouvées `MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native`. Par exemple, pour Windows, la bibliothèque

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

• Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

**Scénario associé**

Ce composant s’utilise de la même manière que le **tHiveLoad**. Pour plus d’informations, consultez Scénario : Créer une table Hive partitionnée à la page 1680.
**tImpalaOutput**

Exécute l’action définie sur les données contenues dans la table, à partir du flux provenant du composant précédent dans le Job.

tImpalaOutput se connecte à une base de données Impala (système d'entrepôt de données Impala) et écrit des données dans une table Impala.

**Propriétés du tImpalaOutput Standard**

Ces propriétés sont utilisées pour configurer le tImpalaOutput s'exécutant dans le framework de Jobs Standard.

Le composant tImpalaOutput Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Built-in</em></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><em>Repository</em></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
</tbody>
</table>

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

**Distribution**

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que
vous utilisez. Les options de la liste dépendent des composants que vous utilisez. Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez *Amazon EMR*, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center ([https://help.talend.com](https://help.talend.com)).

- L’option *Custom* vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner *Import from existing version* pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner *Import from zip* pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste [Hadoop configuration](en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option *Import from existing version*, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d'importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Impala version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez dans ce champ le nom de la base de données.</td>
</tr>
<tr>
<td>Username</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
</tbody>
</table>
| Use kerberos authentication | Si vous accédez à un système Impala s’exécutant avec la sécurité Kerberos, cochez la case et saisissez le Principal Kerberos de ce système Impala.  
  • Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.  
  Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.  
  Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez. |
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
  Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
  • View schema : sélectionnez cette option afin de voir le schéma. |
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

| Table Name | Saisissez le nom de la table dans laquelle écrire les données. |
| Action | Sélectionnez **OVERWRITE** pour écraser les données existantes à l'emplacement de destination ou sélectionnez **APPEND** pour écrire les nouvelles données à la suite des anciennes. |
| Extended insert | Cochez cette case pour combiner de multiples lignes de données en une seule action **INSERT**. Cela permet d’accélérer l’opération d’insertion. |

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**. |

## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes Impala SQL.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  
La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.  
| Prérequis           | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend.  
La liste suivante présente des informations d’exemple relatives à MapR.  
- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).  
Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.  
- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments. |
de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Scénario associé

Pour un scénario concernant l’utilisation d’un composant de sortie dans un Job, consultez Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667.
**tImpalaRow**

Agit sur la structure même de la base de données ou sur les données (mais sans les manipuler).

Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes. tImpalaRow est le composant spécifique à ce type de base de données. Il exécute des requêtes Impala SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

**Propriétés du tImpalaRow Standard**

Ces propriétés sont utilisées pour configurer le tImpalaRow s’exécutant dans le framework de Jobs Standard.

Le composant tImpalaRow Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.
Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez. Parmi ces options, les suivantes nécessitent une configuration spécifique.


- **Si vous sélectionnez Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- **L’option Custom** vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**.

  1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

  2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend**. **Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.
<table>
<thead>
<tr>
<th><strong>Impala version</strong></th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d'écoute du serveur.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez dans ce champ le nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username</strong></td>
<td>Informations d'authentification de l’utilisateur de base de données.</td>
</tr>
</tbody>
</table>
| **Use kerberos authentication** | Si vous accédez à un système Impala s’exécutant avec la sécurité Kerberos, cochez la case et saisissez le Principal Kerberos de ce système Impala.  
  - Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l'authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.  
  Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.  
  Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.  
  Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
  - **View schema** : sélectionnez cette option afin de voir le schéma. |
**• Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

**• Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Nom de la table à traiter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Guess Query</strong></td>
<td>Cliquez sur le bouton <strong>Guess Query</strong> pour générer la requête correspondant au schéma de votre table dans le champ <strong>Query</strong>.</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien <strong>Row &gt; Rejects</strong>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Propagate QUERY’s recordset</th>
<th>Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste <strong>use column</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remarque</strong></td>
<td>Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type <strong>Object</strong>.</td>
</tr>
<tr>
<td>Ce composant est généralement suivi du tParseRecordSet.</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td></td>
</tr>
</tbody>
</table>

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables


### Utilisation

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes Impala SQL. |

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez le Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et le Scénario : Lire des données à partir |
de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

• Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

• Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Scénarios associés

Pour des sujets associés, consultez :

• Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
• Procédure.
• Scénario : Supprimer et re-générer un index de table MySQL à la page 2700.
**tInfiniteLoop**

Exécute automatiquement une tâche ou un Job dans une boucle, à l’infini.

`tInfiniteLoop` fait une boucle à l’infini sur l’exécution d’une tâche.

### Propriétés du `tInfiniteLoop Standard`

Ces propriétés sont utilisées pour configurer le `tInfiniteLoop` s’exécutant dans le framework de Jobs Standard.

Le composant `tInfiniteLoop Standard` appartient à la famille Orchestration.

Le composant de ce framework est toujours disponible.

#### Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait at each iteration (in milliseconds)</td>
<td>Saisissez le temps, en millisecondes, d’attente entre chaque itération.</td>
</tr>
</tbody>
</table>

#### Advanced settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

#### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

#### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Le composant <code>tInfiniteLoop</code> est un composant de début et requiert une connexion de type Iterate au composant suivant.</td>
</tr>
<tr>
<td>Connections</td>
<td>Liens de sortie (de ce composant à un autre) :</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td><strong>Row</strong> : Iterate.</td>
</tr>
<tr>
<td></td>
<td>Liens d’entrée (d’un autre composant à celui-ci) :</td>
</tr>
<tr>
<td></td>
<td><strong>Row</strong> : Main.</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant les liens, consultez le Guide utilisateur du Studio Talend.

### Scénario associé

Pour un exemple d’utilisation proche, consultez Procédure à la page 2110 du composant tLoop.
**tInformixBulkExec**

Exécute des opérations d'Insert dans une base de données Informix.

Les composants tInformixOutputBulk et tInformixBulkExec sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tInformixOutputBulkExec. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

**Propriétés du tInformixBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tInformixBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tInformixBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

*Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Execution Platform</td>
<td>Sélectionnez le type de système d’exploitation que vous utilisez.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

*Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :
1. **Au niveau parent**, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. **Au niveau enfant**, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Nom du schéma.</td>
</tr>
</tbody>
</table>
| **Username et Password** | Informations d’authentification sur l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Instance** | Nom de l’instance Informix à utiliser. Cette information se trouve généralement dans le fichier **SQL hosts**. |
| **Table** | Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois. |
| **Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
**None** : n’effectuer aucune opération de table.  
**Drop and create the table** : supprimer la table puis en créer une nouvelle.  
**Create a table** : créer une table qui n’existe pas encore.  
**Create table if doesn’t exist** : créer la table si nécessaire.  
**Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.  
**Clear table** : supprimer le contenu de la table. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.  
**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**. |
InformixBulkExec


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Informix Directory**

Répertoire d’installation d’Informix, par exemple "C:\Program Files\IBM\IBM Informix Dynamic Server\11.50".

**Data file**

Nom du fichier à charger.

**Action on data**

Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

- **Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
- **Update** : Mettre à jour les entrées existantes.
- **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
- **Delete** : Supprimer les entrées correspondantes au flux d’entrée.

**Avertissement** :

Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations Update et Delete. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une
InformixBulkExec

<table>
<thead>
<tr>
<th>Advanced settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Additional JDBC parameters</strong></td>
</tr>
<tr>
<td><strong>Field terminated by</strong></td>
</tr>
<tr>
<td><strong>Set DBMONEY</strong></td>
</tr>
<tr>
<td><strong>Set DBDATE</strong></td>
</tr>
<tr>
<td><strong>Rows Before Commit</strong></td>
</tr>
<tr>
<td><strong>Bad Rows Before Abort</strong></td>
</tr>
<tr>
<td><strong>tStat Catcher Statistics</strong></td>
</tr>
<tr>
<td><strong>Output</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Global Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global Variables</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes DB2.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.


| Limitation | Le serveur/client de la base de données doit être installé sur la même machine que le Studio Talend ou que le Job contenant un **tInformixBulkExec**, afin que le Job fonctionne correctement.  

Ce composant requiert l’installation des fichiers .jar liés. |

**Scénario associé**

Pour un scénario associé au composant **tInformixBulkExec**, consultez :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du tMysqlOutputBulkExec.
• Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du tOracleBulkExec.
tInformixClose

ferme une connexion à la base de données Informix.

tInformixClose ferme la connexion à une base de données connectée.

**Propriétés du tInformixClose Standard**

Ces propriétés sont utilisées pour configurer le tInformixClose s'exécutant dans le framework de Jobs Standard.

Le composant tInformixClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tInformixConnection dans la liste.</td>
</tr>
</tbody>
</table>

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
</table>
| Dynamic settings   | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exem
ple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénario associé

Ce composant est étroitement lié aux composants tInformixConnection et tInformixRollback. Ils sont généralement utilisés avec un composant tInformixConnection car il permet d’ouvrir une connexion pour la transaction en cours.

Pour un scénario associé au composant tInformixClose, consultez tMysqlConnection à la page 2618.
tInformixCommit

Commite en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes.

Ce composant permet un gain de performance et est étroitement lié aux composants tInformixConnection et tInformixRollback. Ils sont généralement utilisés ensemble lors de transactions.

Le composant tInformixCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonction Auto Commit disponible dans le composant de connexion.

Propriétés du tInformixCommit Standard

Ces propriétés sont utilisées pour configurer le tInformixCommit s’exécutant dans le framework de Jobs Standard.

Le composant tInformixCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

| Database | Sélectionnez un type de base de données dans la liste et cliquez sur Apply. |
| Component list | S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tInformixConnection dans la liste. |
| Close connection | Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. |

Avertissement :

Si vous utilisez un lien de type Row > Main pour relier le tInformixCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.
**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

| Règle d'utilisation | Ce composant est généralement utilisé avec des composants Informix et notamment tInformixConnection et tInformixRollback. |

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez [Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte](#) à la page 2641 et [Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement](#) à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le [Guide utilisateur du Studio Talend](#).

**Scénario associé**

Ce composant est étroitement lié aux composants tInformixConnection et tInformixRollback. Ils sont généralement utilisés avec un composant tInformixConnection car il permet d’ouvrir une connexion pour la transaction en cours.

Pour un scénario associé au composant tInformixCommit, consultez [Scénario : Insérer des données dans les tables mère/fille](#) à la page 2620.
tInformixConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

tInformixConnection est étroitement lié aux composants tInformixCommit et tInformixRollback. Ils sont généralement utilisés avec un composant tInformixConnection car ce dernier permet d’ouvrir une connexion pour la transaction en cours.

Propriétés du tInformixConnection Standard

Ces propriétés sont utilisées pour configurer le tInformixConnection s’exécutant dans le framework de Jobs Standard.

Le composant tInformixConnection Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets</td>
</tr>
</tbody>
</table>
InformixConnection

<table>
<thead>
<tr>
<th>Instance</th>
<th>Nom de l’instance Informix à utiliser. Cette information se trouve généralement dans le fichier SQL hosts.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional JDBC parameters</td>
<td>Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.</td>
</tr>
<tr>
<td>Use or register a shared DB Connection</td>
<td>Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ Shared DB Connection Name qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion à différents niveaux de Jobs, enfants ou parents. Cette option est incompatible avec les options Use dynamic job et Use an independent process to run subjob du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.</td>
</tr>
</tbody>
</table>

Advanced settings

| Use Transaction | Décochez cette case lorsque la base de données est configurée en mode NO_LOG. Si la case est cochée, vous pouvez choisir d’activer ou non l’option Auto Commit. |
| Auto Commit | Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit. Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne committe pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants Informix, notamment les composants tInformixCommit et tInformixRollback.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Ce composant requiert l’installation des fichiers .jar liés.</td>
</tr>
</tbody>
</table>

Scénarios associés

Pour un scénario associé au composant tInformixConnection, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
Ce composant lit une base de données et en extrait des champs à l'aide de requêtes.

Le tInformixInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

**Propriétés du tInformixInput Standard**

Ces propriétés sont utilisées pour configurer le tInformixInput s'exécutant dans le framework de Jobs Standard.

Le composant tInformixInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

| Database | Sélectionnez un type de base de données dans la liste et cliquez sur Apply. |
| Property type | Peut être Built-in ou Repository |
| Built-in | Propriétés utilisées ponctuellement. |
| Repository | Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |
| ![ ] | Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend. |
| Use an existing connection | Cochez cette case et sélectionnez le composant adéquat à partir de la liste Component pour réutiliser les paramètres d’une connexion que vous avez déjà définis. |
| Host name | Adresse IP du serveur de base de données. |
| Port | Numéro de port d’écoute du serveur. |
| **Database** | Nom de la base de données. |
| **Schema** | Nom du schéma. |
| **Username et Password** | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Instance** | Nom de l’instance Informix à utiliser. Cette information se trouve généralement dans le fichier SQL hosts. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
**Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• **View schema** : sélectionnez cette option afin de voir le schéma.  
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| **Query type et Query** | Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma. |
| **Guess Query** | Cliquez sur le bouton Guess Query pour générer la requête correspondant au schéma de votre table dans le champ Query. |
| **Guess schema** | Cliquez sur le bouton pour récupérer le schéma de la table. |
### Advanced settings

<table>
<thead>
<tr>
<th><strong>Advanced settings</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Additional JDBC parameters</strong></td>
<td>Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case <strong>Use an existing connection</strong> est décochée dans les <strong>Basic settings</strong>.</td>
</tr>
<tr>
<td><strong>Trim all the String/Char columns</strong></td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td><strong>Trim column</strong></td>
<td>Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th><strong>Global Variables</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| **Global Variables** | NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.  
QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.  
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

<table>
<thead>
<tr>
<th><strong>Règle d’utilisation</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Règle d’utilisation</strong></td>
<td>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Informix.</td>
</tr>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exem</td>
</tr>
</tbody>
</table>
Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

| Limitation | Ce composant requiert l’installation des fichiers .jar liés. |

**Scénarios associés**

Consultez les scénarios associés dans :

Consultez également **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520.
## tInformixOutput

Exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

*InformixOutput écrit, met à jour, modifie ou supprime les données d’une base de données.*

### Propriétés du tInformixOutput Standard

Ces propriétés sont utilisées pour configurer le *InformixOutput* s’exécutant dans le framework de Jobs Standard.

Le composant *InformixOutput* Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez [Composants de bases de données dynamiques](#) à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong> :</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>![ ]</td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue <strong>Basic settings</strong> du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>![ ]</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie. <strong>Remarque :</strong> Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par...</td>
</tr>
</tbody>
</table>
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

| **Host** | Adresse IP du serveur de base de données. |
| **Port** | Numéro de port d’écoute du serveur. |
| **Database** | Nom de la base de données. |
| **Schema** | Nom du schéma. |
| **Username et Password** | Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Instance** | Nom de l’instance Informix à utiliser. Cette information se trouve généralement dans le fichier **SQL hosts**. |
| **Table** | Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois. |
| **Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :
  - **None** : n’effectuer aucune opération de table.
  - **Drop and create the table** : supprimer la table puis en créer une nouvelle.
  - **Create a table** : créer une table qui n’existe pas encore.
  - **Create table if doesn’t exist** : créer la table si nécessaire.
  - **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.
  - **Clear a table** : supprimer le contenu de la table.
  - **Truncate table** : tronquer la table.
  **Avertissement** :
  Une opération de commit est effectuée après troncature de la table. |
| **Action on data** | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : |
**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

**Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.

**Update** : Mettre à jour les entrées existantes.

**Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.

**Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré.

**Delete** : Supprimer les entrées correspondantes au flux d'entrée.

⚠️ **Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update et Delete**. Pour cela, cliquez sur le bouton `[...]` à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l'onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).
- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects.

**Advanced settings**

<table>
<thead>
<tr>
<th>Additional JDBC parameters</th>
<th>Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case Use an existing connection est décochée dans les Basic settings.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque</strong> :</td>
</tr>
<tr>
<td></td>
<td>Vous pouvez appuyer sur Ctrl+Espace afin d’accéder à une liste de variables globales prédéfinies.</td>
</tr>
<tr>
<td>Commit every</td>
<td>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</td>
</tr>
<tr>
<td>Additional Columns</td>
<td>Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</td>
</tr>
<tr>
<td></td>
<td><strong>Name</strong> : Saisissez le nom de la colonne à modifier ou à insérer.</td>
</tr>
<tr>
<td></td>
<td><strong>SQL expression</strong> : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.</td>
</tr>
<tr>
<td></td>
<td><strong>Position</strong> : Sélectionnez Before, Replace ou After, en fonction de l’action à effectuer sur la colonne de référence.</td>
</tr>
</tbody>
</table>
### Reference column
Saisissez une colonne de référence que le composant `tInformixOutput` peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

<table>
<thead>
<tr>
<th>Use field options</th>
<th>Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable debug mode</td>
<td>Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.</td>
</tr>
<tr>
<td>Use Batch</td>
<td>Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.</td>
</tr>
<tr>
<td>Batch Size</td>
<td>Spécifiez le nombre d’enregistrements à traiter dans chaque lot. Ce champ est disponible uniquement lorsque la case <strong>Use batch mode</strong> est cochée.</td>
</tr>
<tr>
<td>Optimize the batch insertion</td>
<td>Cocher cette case optimise l’insertion de données par lots.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong></td>
<td>nombre de lignes mises à jour. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong></td>
<td>nombre de lignes insérées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong></td>
<td>nombre de lignes supprimées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_REJECTED</strong></td>
<td>nombre de lignes rejetées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>QUERY</strong></td>
<td>requête traitée. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette...
Vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Informix. Il permet aussi de créer un flux de rejet avec un lien Row &gt; Reject filtrant les données en erreur. Pour un exemple d'utilisation, consultez Scénario : Récupérer les données erronées à l'aide d’un lien Reject à la page 2675 du composant tMysqlOutput.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

| Limitation          | Ce composant requiert l'installation des fichiers .jar liés. |

### Scénarios associés

Pour un scénario associé au composant tInformixOutput, consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667 du composant tMysqlOutput.
**tInformixOutputBulk**

Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Informix.

Les composants tInformixOutputBulk et tInformixBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tInformixOutputBulkExec. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Écrit un fichier composé de colonnes et basé sur le séparateur défini et sur les standards Informix.

**Propriétés du tInformixOutputBulk Standard**

Ces propriétés sont utilisées pour configurer le tInformixOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tInformixOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>File Name</strong></td>
<td>Nom du fichier à générer.</td>
</tr>
<tr>
<td><strong>Append</strong></td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs.</td>
</tr>
</tbody>
</table>
Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Advanced settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row separator</strong></td>
<td>Chaîne (ex : &quot;\n&quot; sous Unix) séparant les lignes.</td>
</tr>
<tr>
<td><strong>Field separator</strong></td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td><strong>Set DBMONEY</strong></td>
<td>Quand vous cochez cette case, vous pouvez définir votre séparateur decimal dans le champ Decimal separator.</td>
</tr>
<tr>
<td><strong>Set DBDATE</strong></td>
<td>Sélectionnez le format de date que vous souhaitez appliquer.</td>
</tr>
<tr>
<td><strong>Create directory if not exists</strong></td>
<td>Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s’il n’existe pas déjà.</td>
</tr>
<tr>
<td><strong>Custom the flush buffer size</strong></td>
<td>Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ Row number, saisissez le</td>
</tr>
</tbody>
</table>
InformixOutputBulk

nombre de lignes après lesquelles la mémoire est à nouveau libérée.

Encoding

 Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

StatCatcher Statistics

 Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

Global Variables

| NB_LINE | nombre de lignes traitées. Cette variable est une variable After et retourne un entier. |
| ERROR_MESSAGE | message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant est généralement utilisé avec le composant tInformixBulkExec. Ensemble, ils offrent un gain de performance important pour l’alimentation d’une base de données Informix.

Dynamic settings

Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.
Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>Limitation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
<td></td>
</tr>
</tbody>
</table>

### Scénarios associés

Pour un scénario associé au **tInformixOutputBulk**, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du composant **tMysqlOutputBulk**.
- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du **tMysqlOutputBulkExec**.
tInformixOutputBulkExec

Effectue des opérations d’Insert sur les données fournies dans une base de données Informix.

Les composants tInformixOutputBulk et tInformixBulkExec sont généralement utilisés ensemble comme deux parties d’un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l’opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tInformixOutputBulkExec.

Propriétés du tInformixOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tInformixOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tInformixOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque** :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

| **Database** | Sélectionnez un type de base de données dans la liste et cliquez sur **Apply**. |
| **Property Type** | Peut être **Built-in** ou **Repository**. |
| **Built-in** : Propriétés utilisées ponctuellement. |
| **Repository** : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |
| **Execution platform** | Sélectionnez le type de système d’exploitation que vous utilisez. |
| **Use an existing connection** | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |

**Remarque** :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d'authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Instance</td>
<td>Nom de l’instance Informix à utiliser. Cette information se trouve généralement dans le fichier SQL hosts.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit déjà exister pour que l’opération d’insert soit autorisée.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- View schema : selectionnez cette option afin de voir le schéma.
- Change to built-in property : selectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : selectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et selectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Informix Directory
Répertoire d’installation d’Informix, par exemple "C:\Program Files\IBM\IBM Informix Dynamic Server\11.50\".

Data file
Nom du fichier à générer et à charger.

Append
Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.

Action on data
Sélectionnez l’opération que vous voulez effectuer :
Bulk insert ou Bulk update. Les informations demandées seront différentes en fonction de l’action choisie.

Advanced settings

Additional JDBC parameters
Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case Use an existing connection est décochée dans les Basic settings.

Remarque :
Vous pouvez appuyer sur Ctrl+Espace afin d’accéder à une liste de variables globales prédéfinies.
<table>
<thead>
<tr>
<th>Row separator</th>
<th>Chaîne (ex : <code>\n</code> sous Unix) séparant les lignes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields terminated by</td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td>Set DBMONEY</td>
<td>Quand vous cochez cette case, vous pouvez définir votre séparateur décimal dans le champ <em>Decimal separator</em>.</td>
</tr>
<tr>
<td>Set DBDATE</td>
<td>Sélectionnez le format de date que vous souhaitez appliquer.</td>
</tr>
<tr>
<td>Rows Before Commit</td>
<td>Saisissez le nombre de lignes à traiter avant commit.</td>
</tr>
<tr>
<td>Bad Rows Before Abort</td>
<td>Saisissez le nombre de lignes d’erreur avant arrêt du Job.</td>
</tr>
<tr>
<td>Create directory if not exists</td>
<td>Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s’il n’existe pas déjà.</td>
</tr>
<tr>
<td>Custom the flush buffer size</td>
<td>Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ <em>Row number</em>, saisissez le nombre de lignes après lesquelles la mémoire est à nouveau libérée.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez <em>Custom</em> et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
<tr>
<td>Output</td>
<td>Sélectionnez l’emplacement de la sortie.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton <code>[+]</code> pour ajouter une ligne à la table. Dans le champ <em>Code</em>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste <em>Component List</em> de la vue <em>Basic settings</em> devient inutilisable.</td>
</tr>
</tbody>
</table>
Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

**Limitation**

Le serveur/client de la base de données doit être installé sur la même machine que le Studio Talend ou le Job contenant un tInformixOutputBulkExec, afin que le composant fonctionne correctement.

### Scénarios associés

Pour un scénario associé au tInformixOutputBulkExec, consultez :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du composant tMysqlOutputBulk.
- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec.
**tInformixRollback**

Evite le commit de transaction involontaire en annulant la transaction dans une base de données connectée.

tInformixRollback est étroitement lié aux composants tInformixCommit et tInformixConnection. Ils sont généralement utilisés ensemble lors de transactions.

**Propriétés du tInformixRollback Standard**

Ces propriétés sont utilisées pour configurer le tInformixRollback s’exécutant dans le framework de Jobs Standard.

Le composant tInformixRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

![i] Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

| Database | Sélectionnez un type de base de données dans la liste et cliquez sur Apply. |
| Component list | Sélectionnez le composant de connexion tInformixConnection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours. |
| Close Connection | Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

| Règle d’utilisation | Il faut utiliser ce composant en association avec des composants Informix, notamment avec le tInformixConnection et le tInformixCommit. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans |
Votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

Lorsqu'un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Scénario associé

Pour un scénario associé au tInformixRollback, consultez Scénario : Annuler l'insertion de données dans des tables mère/fille à la page 2623 du composant tMysqlRollback.
**tInformixRow**

Agit sur la structure même de la base de données ou sur les données (mais sans les manipuler) grâce au SQLBuilder qui aide à écrire les requêtes.

tInformixRow exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

**Propriétés du tInformixRow Standard**

Ces propriétés sont utilisées pour configurer le tInformixRow s’exécutant dans le framework de Jobs Standard.

Le composant tInformixRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
la vue **Basic settings** du composant de connexion créant cette connexion.

2. **Au niveau enfant,** utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend.**

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password,** puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| Instance | Nom de l’instance Informix à utiliser. Cette information se trouve généralement dans le fichier **SQL hosts.** |
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs. |
| Built-in | Le schéma est créé et conservé pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend.** |
| Repository | Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend.** |
| Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository,** trois options sont disponibles : |
| • **View schema** : sélectionnez cette option afin de voir le schéma. |
| • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. |
| • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job |
InformixRow

1847

courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

| Table Name | Nom de la table. |
| Query type | La requête peut être Built-in ou distante dans le Repository |
| Built-in | Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder. |
| Repository | Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement. |
| Query | Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma. |
| Guess Query | Cliquez sur le bouton Guess Query pour générer la requête correspondant au schéma de votre table dans le champ Query. |
| Die on error | Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects. |
| Use Transaction | Décochez cette case lorsque la base de données est configurée en mode NO_LOG. |

Advanced settings

| Additional JDBC parameters | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case Use an existing connection est décochée dans les Basic settings. |
| Propagate QUERY’s recordset | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column. |

Remarque :
Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type Object. Ce composant est généralement suivi du tParseRecordSet.

| Use PreparedStatement | Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau Set PreparedStatement Parameter, définissez les valeurs des paramètres |
représentés par des ‘?’ dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**.

**Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL.

**Parameter Type** : Saisissez le type du paramètre.

**Parameter Value** : Saisissez la valeur du paramètre.

**Remarque** :
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

<table>
<thead>
<tr>
<th>Commit every</th>
<th>Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l'exécution d’un composant. Une variable After fonctionne après l'exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez</td>
</tr>
</tbody>
</table>
informixRow

dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


| Limitation | Ce composant requiert l’installation des fichiers .jar liés. |

Scénarios associés

Pour un scénario associé, consultez :

- Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
- Procédure du composant tDBSQLRow.
- Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du composant tMysqlRow.
tInformixSCD

Reflète et traque les modifications d’une table Informix SCD dédiée.

`tInformixSCD` répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.

**Propriétés du tInformixSCD Standard**

Ces propriétés sont utilisées pour configurer le tInformixSCD s’exécutant dans le framework de Jobs Standard.

Le composant tInformixSCD Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique.

Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Nom du schéma de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Instance</strong></td>
<td>Nom de l’instance Informix à utiliser. Cette information se trouve généralement dans le fichier SQL hosts.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
</tbody>
</table>
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| **Built-in** | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*. |
InformixSCD


### SCD Editor

L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension.

Pour plus d’informations, consultez Méthodologie de gestion du SCD à la page 2716.

### Use memory saving Mode

Cochez cette case pour améliorer les performances du système.

### Source keys include Null

Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null.

⚠️ **Avertissement** :

Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.

### Use Transaction

Décochez cette case lorsque la base de données est configurée en mode NO_LOG.

### Die on error

Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

### Advanced settings

#### End date time details

Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format HH:mm:ss. La valeur par défaut pour ce champ est 12:00:00.

Ce champ apparaît uniquement lorsqu’un SCD de Type 2 est utilisé et lorsque Fixed year value est sélectionné pour créer la date de fin du SCD.

#### Debug mode

Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.

#### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

### Variables globales

#### Global Variables

- **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.

- **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.

- **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaine de caractères.
**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Limitation</td>
<td>Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez tMysqlSCD à la page 2712.
**tInformixSP**

Centralise et appelle des requêtes multiples ou complexes dans une base de données. Le composant tInformixSP appelle une procédure stockée de base de données.

**Propriétés du tInformixSP Standard**

Ces propriétés sont utilisées pour configurer le tInformixSP s’exécutant dans le framework de Jobs Standard.

Le composant tInformixSP Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>
| **Remarque :** | Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. |
<table>
<thead>
<tr>
<th><strong>InformixSP</strong></th>
<th>Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le <em>Guide utilisateur du Studio Talend</em>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro de port d’écoute du serveur.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <em>Password</em>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Instance</strong></td>
<td>Nom de l’instance Informix à utiliser. Cette information se trouve généralement dans le fichier <em>SQL hosts</em>.</td>
</tr>
<tr>
<td><strong>SP Name</strong></td>
<td>Saisissez le nom exact de la Procédure Stockée (SP).</td>
</tr>
<tr>
<td><strong>Is Function / Return result in</strong></td>
<td>Cochez cette case si une seule valeur doit être retournée. Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>Cliquez sur le bouton [+] et sélectionnez dans le champ <strong>Schema Columns</strong> les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure. Sélectionnez le <strong>Type</strong> de paramètre : <strong>IN</strong> : paramètre d'entrée (Input). <strong>OUT</strong> : paramètre de sortie (Output)/valeur retournée. <strong>IN OUT</strong> : les paramètres d'entrée doivent être retournées sous forme de valeur, même après modifications via la procédure (fonction). <strong>RECORDSET</strong> : les paramètres d'entrée doivent être retournées sous forme d'ensemble de valeurs, au lieu d'une valeur unique. <strong>Remarque</strong> : Consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620 si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d'une requête SQL.</td>
</tr>
<tr>
<td><strong>Use Transaction</strong></td>
<td>Décochez cette case lorsque la base de données est configurée en mode NO_LOG.</td>
</tr>
<tr>
<td><strong>Advanced settings</strong></td>
<td><strong>Additional JDBC parameters</strong> Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case <strong>Use an existing connection</strong> est décochée dans les <strong>Basic settings</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>tStatCatcher Statistics</strong> Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
<tr>
<td><strong>Utilisation</strong></td>
<td><strong>Règle d’utilisation</strong> Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d’entrée sont autorisés. <strong>Dynamic settings</strong> Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez</td>
</tr>
</tbody>
</table>
informixSP dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


| Limitation | La syntaxe de la Procédure Stockée doit correspondre à celle de la base de données. Ce composant requiert l’installation des fichiers .jar liés. |

**Scénarios associés**

Pour un scénario associé, consultez :

- Scénario : Récupérer des informations personnelles à l’aide d’une procédure stockée à la page 2594.
- Scénario : Utiliser le tMysqlSP pour trouver le libellé State à l’aide d’une procédure stockée à la page 2734.
- Scénario : Vérifier le format de numéros à l’aide d’une procédure stockée à la page 2976.
- Scénario : Exécuter une procédure stockée à l’aide du tMDMSP à la page 2320.

Consultez également Scénario : Insérer des données dans des tables mère/fille à la page 2620 si vous voulez analyser un ensemble d’enregistrements d’une table de données ou d’une requête SQL.
tIngresBulkExec

Insère des données en masse dans une table du SGBD Ingres afin d’optimiser les performances.

Les composants tIngresOutputBulk et tIngresBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tIngresOutputBulkExec. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Propriétés du tIngresBulkExec Standard

Ces propriétés sont utilisées pour configurer le tIngresBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tIngresBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à remplir.</td>
</tr>
<tr>
<td>VNode</td>
<td>Nom du nœud virtuel.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Actions pouvant être effectuées sur la table :</td>
</tr>
<tr>
<td>None</td>
<td>Aucune opération.</td>
</tr>
<tr>
<td>Truncate</td>
<td>Supprime toutes les lignes de la table et rend l’espace du fichier au système d’exploitation.</td>
</tr>
<tr>
<td>File name</td>
<td>Nom du fichier à charger.</td>
</tr>
</tbody>
</table>
### Avertissement :

Ce fichier doit se situer sur la même machine que le serveur de la base de données.

<table>
<thead>
<tr>
<th><strong>Schema et Edit Schema</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <code>line</code> lors du nommage des champs.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Built-In</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Repository</strong></th>
</tr>
</thead>
</table>
| Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. 
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets. 
Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com). |

<table>
<thead>
<tr>
<th><strong>Delete Working Files After Use</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour supprimer les fichiers créés durant l'exécution.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Field Separator</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisissez le caractère, la chaîne de caractères ou l'expression régulière permettant de séparer les champs.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Row Separator</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Chaîne de caractères (ex : &quot;\n&quot; sous Unix) permettant de séparer les lignes.</td>
</tr>
<tr>
<td>Null Indicator</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>Session User</td>
</tr>
<tr>
<td>Rollback</td>
</tr>
<tr>
<td>On Error</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Reject Row File</td>
</tr>
<tr>
<td>Error Count</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Allocation</td>
</tr>
<tr>
<td>Extend</td>
</tr>
<tr>
<td>Fill Factor</td>
</tr>
<tr>
<td>Min Pages/Max Pages</td>
</tr>
<tr>
<td>Leaf Fill</td>
</tr>
<tr>
<td>Non Leaf Fill</td>
</tr>
<tr>
<td>Row Estimate</td>
</tr>
<tr>
<td>Trailing WhiteSpace</td>
</tr>
</tbody>
</table>
IngresBulkExec

Output

Emplacement où écrire le message d’erreur :

- **to console** : Ecrire le message dans la console.
- **to global variable** : Ecrire le message dans une variable globale.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE_DATA : nombre de lignes lues. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_BAD : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

**Utilisation**

**Règle d’utilisation**

Le composant tIngresBulkExec est généralement utilisé avec le tIngresConnection et le tIngresRow. Il permet d’alimenter en masse une base de données Ingres, afin d’optimiser les performances.

**Limitation**

Le serveur/client de la base de données doit être installé sur la même machine que le **Studio Talend** ou que le Job contenant un tIngresBulkExec, afin que le composant fonctionne correctement.

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).
Scénario associé

Pour un scénario associé, consultez :

• Scénario : Charger des données dans une table du SGBD Ingres à la page 1886.
**tIngresClose**

Ferme la connexion à une base de données Ingres connectée.

**Propriétés du tIngresClose Standard**

Ces propriétés sont utilisées pour configurer le tIngresClose s'exécutant dans le framework de Jobs Standard.

Le composant tIngresClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <em>Apply</em>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S'il y a plus d’une connexion dans le Job en cours, sélectionnez le composant <em>tIngresConnection</em> dans la liste.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie. |
| Dynamic settings | Cliquez sur le bouton [*+] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend. |
Lorsqu'un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tIngresCommit**

Ce composant commite en une fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes, ce qui permet un gain de performance.

Le tIngresCommit valide les données traitées dans un Job à partir d’une base de données connectée.

**Propriétés du tIngresCommit Standard**

Ces propriétés sont utilisées pour configurer le tIngresCommit s’exécutant dans le framework de Jobs Standard.

Le composant tIngresCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tIngresConnection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Avertissement :**

Si vous utilisez un lien de type Row > Main pour relier le tIngresCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants JDBC et notamment avec les composants <em>tIngresConnection</em> et <em>tIngresRollback</em>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Scénario associé

Pour un scénario associé au *tIngresCommit*, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
**tIngresConnection**

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

**IngresConnection** ouvre une connexion vers une base de données afin d’effectuer une transaction.

**Propriétés du tIngresConnection Standard**

Ces propriétés sont utilisées pour configurer le tIngresConnection s’exécutant dans le framework de Jobs Standard.

Le composant tIngresConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th><strong>Database</strong></th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Server</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Table Schema</strong></td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
### Use or register a shared DB Connection

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un composant **tRunJob** ayant une de ces options activée fera échouer votre Job.

### Advanced settings

#### Auto Commit

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

#### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu’au niveau de chaque composant.

### Utilisation

#### Règle d’utilisation

Ce composant est généralement utilisé avec des composants JDBC et notamment avec les composants **tingresCommit** et **tingresRollback**.

#### Limitation

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez
Scénario associé

Pour un scénario associé au tIngresConnection, consultez Scénario : Charger des données dans une table du SGBD Ingres à la page 1886.
tIngresInput

Lit une base de données Ingres et en extrait des champs à l’aide de requêtes.

tIngresInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

Propriétés du tIngresInput Standard

Ces propriétés sont utilisées pour configurer le tIngresInput s’exécutant dans le framework de Jobs Standard.

Le composant tIngresInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>![icon] Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue <strong>Basic settings</strong> du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
<td></td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>![icon] Remarque : Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une</td>
<td></td>
</tr>
</tbody>
</table>
connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

### Server
Adresse IP du serveur de base de données.

### Port
Numéro du port d’écoute du serveur de base de données.

### Database
Nom de la base de données.

### Username et Password
Informations d’authentification de l’utilisateur de base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

### Schema et Edit Schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

**Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le **Guide utilisateur de Studio Talend**.

**Repository** : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur de Studio Talend**.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez
propérer les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Query type et Query**

Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

**Advanced settings**

**Trim all the String/Char columns**

Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.

**Trim column**

Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

- **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.
- **QUERY** : requête traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

**Utilisation**

**Règle d’utilisation**

Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Ingres.

**Limitation**

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les

### Scénarios associés

Consultez les scénarios associés :

Consultez également Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520.
**tIngresOutput**

Exécute l’action définie sur la table et/ou sur les données d’une table en fonction du flux entrant provenant du composant précédent.

tIngresOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

**Propriétés du tIngresOutput Standard**

Ces propriétés sont utilisées pour configurer le tIngresOutput s’exécutant dans le framework de Jobs Standard.

Le composant tIngresOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par...
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

| **Host** | Adresse IP du serveur de base de données. |
| **Port** | Numéro de port d’écoute du serveur. |
| **Database** | Nom de la base de données. |
| **Username et Password** | Informations d’authentification de l’utilisateur de base de données. 
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Table** | Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois. |
| **Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
- **None** : n’effectuer aucune opération de table.  
- **Drop and create the table** : supprimer la table puis en créer une nouvelle.  
- **Create a table** : créer une table qui n’existe pas encore.  
- **Create table if doesn’t exist** : créer la table si nécessaire.  
- **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.  
- **Clear a table** : supprimer le contenu de la table. |
| **Action on data** | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :  
- **Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.  
- **Update** : Mettre à jour les entrées existantes.  
- **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. |
**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

**Delete** : Supprimer les entrées correspondantes au flux d’entrée.

**Avertissement** :
Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center ([https://help.talend.com](https://help.talend.com)).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
### Update repository connection

Sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th><strong>Die on error</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien <strong>Row &gt; Rejects</strong>.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Commit every</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Additional Columns</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisissez le nom de la colonne à modifier ou à insérer.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SQL expression</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Position</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez <strong>Before</strong>, <strong>Replace</strong> ou <strong>After</strong>, en fonction de l’action à effectuer sur la colonne de référence.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Reference column</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisissez une colonne de référence que le composant <strong>tingresOutput</strong> peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Use field options</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Enable debug mode</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Use Batch</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.</td>
</tr>
</tbody>
</table>

**Remarque :**
Cette case est disponible lorsque vous sélectionnez **Insert**, **Update**, ou **Delete** dans la liste **Action on data**.

<table>
<thead>
<tr>
<th>Batch Size</th>
<th>Spécifiez le nombre d’enregistrements à traiter dans chaque lot. Ce champ est disponible uniquement lorsque la case <strong>Use batch mode</strong> est cochée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>QUERY : requête traitée. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données Ingres. Il permet aussi de créer un flux de rejet avec un lien <strong>Row &gt; Reject</strong> filtrant les données en erreur. Pour</th>
</tr>
</thead>
</table>
un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.


**Scénarios associés**

Pour un scénario associé, consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667 du composant tMysqlOutput.
tIngresOutputBulk

Ce composant prépare le fichier dont les données sont insérées en masse dans le SGBD Ingres afin d’optimiser les performances.

Les composants tIngresOutputBulk et tIngresBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie un fichier et d’autre part utiliser ce fichier lors de l’opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tIngresOutputBulkExec.

tIngresOutputBulk prépare un fichier avec le schéma défini et les données provenant du composant précédent.

Propriétés du tIngresOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tIngresOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tIngresOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom du fichier à générer.</td>
</tr>
<tr>
<td></td>
<td>Avertissement :</td>
</tr>
<tr>
<td></td>
<td>Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.</td>
</tr>
<tr>
<td>Append the File</td>
<td>Cochez cette case pour ajouter de nouvelles lignes à la fin du fichier.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark,</td>
</tr>
<tr>
<td><strong>évitez le mot réservé line lors du n</strong>ommage des champs.</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <em>Guide utilisateur du Studio Talend</em>.</td>
<td></td>
</tr>
</tbody>
</table>
| **Repository** : Le schéma existe déjà et il est stocké dans le *Repository*. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.  
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.  
Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com). |
| Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre *Repository Content*. |

**Advanced settings**

<table>
<thead>
<tr>
<th>Field Separator</th>
<th>Saisissez le caractère, la chaîne de caractères ou l’expression régulière permettant de séparer les champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Separator</td>
<td>Chaîne de caractères (ex : &quot;n&quot; sous Unix) permettant de séparer les lignes.</td>
</tr>
<tr>
<td>Include Header</td>
<td>Cochez cette case pour inclure l’en-tête de colonnes dans le fichier.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez <em>Custom</em> et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>
Global Variables

| Global Variables | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

| Règle d’utilisation | Le composant **tIngresOutputBulk** est généralement utilisé avec le **tIngresBulkExec**, afin de sauvegarder des données entrantes dans un fichier. Ces données sont ensuite insérées en masse dans une base de données Ingres **tIngresBulkExec** afin d’optimiser les performances.

| Limitation | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (**https://help.talend.com**).

Scénario associé

Pour un scénario associé, consultez :

- Scénario : Charger des données dans une table du SGBD Ingres à la page 1886,
**tIngresOutputBulkExec**

Insère des données en masse dans le SGBD Ingres afin d’optimiser les performances.

Les composants tIngresOutputBulk et tIngresBulkExec sont généralement utilisés ensemble dans un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l’opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tIngresOutputBulkExec.

tIngresOutputBulkExec prépare un fichier de sortie et l’utilise pour alimenter une table dans le SGBD Ingres.

**Propriétés du tIngresOutputBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tIngresOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tIngresOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez **Composants de bases de données dynamiques** à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Nom de la table à remplir.</td>
</tr>
<tr>
<td><strong>VNode</strong></td>
<td>Nom du nœud virtuel. Le serveur de la base de données doit être installé sur la même machine que le <strong>Studio Talend</strong> ou que le Job contenant un <strong>tIngresOutputBulkExec</strong>.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Action on table</strong></td>
<td>Actions pouvant être effectuées sur la table : <strong>None</strong> : Aucune opération.</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>Supprime toutes les lignes de la table et rend l’espace du fichier au système d’exploitation.</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>File name</strong></td>
<td>Nom du fichier à générer et à charger. <strong>Avertissement :</strong> Ce fichier est généré sur la machine spécifiée dans le champ <strong>VNode</strong> et doit être sur la même machine que le serveur de la base de données.</td>
</tr>
</tbody>
</table>
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. **Repository** : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets. Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com). Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
  - **View schema** : sélectionnez cette option afin de voir le schéma.
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **Repository Content**. |
| **Delete Working Files After Use** | Cochez cette case pour supprimer les fichiers créés durant l’exécution. |
### Advanced settings

<table>
<thead>
<tr>
<th>Field Separator</th>
<th>Saisissez le caractère, la chaîne de caractères ou l'expression régulière permettant de séparer les champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Separator</td>
<td>Chaîne de caractères (ex : &quot;\n&quot; sous Unix) permettant de séparer les lignes.</td>
</tr>
</tbody>
</table>
| On Error        | Politique de gestion d’erreur :  
  - **Continue** : Continuer l’exécution.  
  - **Terminate** : Terminer l’exécution. |
| Reject Row File | Chemin d’accès et nom du fichier contenant les lignes rejetées.  
Disponible lorsque l’option **Continue** est sélectionnée dans la liste **On Error**. |
| Error Count     | Nombre d’erreurs avant arrêt de l’exécution.  
Disponible lorsque l’option **Terminate** est sélectionnée dans la liste **On Error**. |
| Rollback        | Activer ou désactiver le rollback. |
| Null Indicator  | Valeur de l’indicateur null. |
| Session User    | Utilisateur de la session définie (connexion à la base de données). |
| Allocation      | Nombre de pages initialement allouées à la table ou à l’index. |
| Extend          | Nombre de pages étendant la table ou l’index. |
| Fill Factor     | Spécifiez le pourcentage (de 1 à 100) de chaque page principale de données devant être remplie de lignes, dans des conditions idéales. Par exemple, si vous spécifiez un facteur de remplissage de 40, le serveur de SGBD remplit de lignes 40% de chaque page principale dans la table restructurée. |
| Min Pages/Max Pages | Spécifiez le nombre minimum/maximum de pages principales qu’une table de hachage doit posséder. Ces nombres doivent être de 1 minimum. |
| Leaf Fill       | Une copie de masse peut spécifier une valeur leaffill. Cette clause spécifie le pourcentage (de 1 à 100) de chaque index de page de table B-tree devant être rempli par des lignes durant la copie. Cette clause peut être utilisée uniquement sur des tables ayant une structure de stockage B-tree. |
| Non Leaf Fill   | Une copie de masse peut spécifier une valeur nonleaffill. Cette clause spécifie le pourcentage (de 1 à 100) de chaque index de page de table B-tree ne devant pas être rempli par des lignes durant la copie. Cette clause |
peut être utilisée uniquement sur des tables ayant une structure de stockage B-tree.

| Row Estimate | Spécifiez le nombre estimé de lignes à copier d’un fichier à une table durant l’opération de copie de masse. |
| Trailing WhiteSpace | Cochée par défaut, cette case est conçue pour supprimer les espaces en fin de champ et s’applique uniquement aux types de données VARCHAR, NVARCHAR et TEXT. |
| Encoding | Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données. |
| Output | Emplacement où écrire le message d’erreur : to console : Ecrire le message dans la console. to global variable : Ecrire le message dans une variable globale. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

| Règle d’utilisation | Le composant tIngresOutputBulkExec est généralement utilisé avec le tIngresConnection et le tIngresRow. Il prépare un fichier de sortis et l’alimente en masse avec des données dans le SGBD Ingres, afin d’optimiser les performances. |

**Scénario : Charger des données dans une table du SGBD Ingres**

Dans ce scénario, un composant tIngresOutputBulkExec est utilisé afin de préparer un fichier de sortie contenant des données provenant d’un fichier .csv concernant des employés et utilise ensuite ce fichier de sortie afin d’alimenter une table dans une base de données Ingres.
Déposer et relier les composants

Procédure

1. Déposez un tIngresConnection, un tFileInputDelimited et un tIngresOutputBulkExec de la Palette dans l’espace de modélisation graphique.
2. Renommez le tIngresOutputBulkExec en save_a_copy_and_load_to_DB.
3. Reliez le tIngresConnection au tFileInputDelimited à l’aide d’un lien OnSubjobOk.
4. Reliez le tFileInputDelimited au tIngresOutputBulkExec à l’aide d’un lien Row > Main.

Configurer les composants

Procédure

1. Double-cliquez sur le tIngresConnection pour ouvrir sa vue Component.

![Component view](image)

2. Dans le champ Server, saisissez l'adresse du serveur où se trouve le SGBD Ingres, par exemple "localhost".
   Laissez la valeur du champ Port par défaut.
3. Dans le champ Database, saisissez le nom de la base de données Ingres, par exemple "research".
4. Dans les champs Username et Password, saisissez les informations d'authentification.
   Une variable de contexte est utilisée pour le mot de passe. Pour plus d’informations concernant les variables de contexte, consultez le Guide utilisateur du Studio Talend.
5. Double-cliquez sur le composant tFileInputDelimited pour ouvrir sa vue Component.

7. Cliquez sur le bouton [...] à côté du champ Edit schema afin d'ouvrir l'éditeur de schéma.

8. Cliquez quatre fois sur le bouton [+] pour ajouter quatre colonnes, par exemple name, age, job et dept, dont le type de données est respectivement string, integer, string et string.
   Cliquez sur OK pour fermer l'éditeur de schéma.
   Cliquez sur Yes dans la fenêtre qui s'ouvre afin d'accepter la propagation des modifications au composant suivant.
   Laissez les autres composants tels qu’ils sont.

Dans le champ **Table**, saisissez le nom de la table pour l’insertion de données.

Dans les champs **VNode** et **Database**, saisissez le nom du nœud virtuel et de la base de données.

Dans le champ **File Name**, saisissez le chemin d’accès complet au fichier qui contiendra les données du fichier source.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** pour exécuter le Job.

Comme affiché ci-dessus, les données concernant les employés sont écrites dans la table *employee* dans la base de données *research*, sur le nœud *talendbj*. Le fichier de sortie *employee_research.csv* a été généré à l’emplacement spécifié, *C:/Users/talend/Desktop*.

**Scénarios associés**

Pour des scénarios associés, consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667.
**tIngresRollback**

Evite le commit de transaction involontaire en annulant la transaction dans une base de données connectée.

**Propriétés du tIngresRollback Standard**

Ces propriétés sont utilisées pour configurer le tIngresRollback s'exécutant dans le framework de Jobs Standard.

Le composant tIngresRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant de connexion <strong>tIngresConnection</strong> dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décrochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est généralement utilisé en association avec des composants Ingres, notamment avec le <strong>tIngresConnection</strong> et le <strong>tIngresCommit</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez...</td>
</tr>
</tbody>
</table>
dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénario associé

Pour un scénario associé au tIngresRollback, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623 du composant tMysqlRollback.
## tIngresRow

Agit sur la structure même de la base de données ou sur les données (mais sans les manipuler) en utilisant le SQLBuilder pour écrire rapidement et aisément les requêtes.

`tIngresRow` exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

### Propriétés du tIngresRow Standard

Ces propriétés sont utilisées pour configurer le `tIngresRow` s'exécutant dans le framework de Jobs Standard.

Le composant `tIngresRow` Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

#### Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez [Composants de bases de données dynamiques](#) à la page 634.

#### Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <em>Apply</em>.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <em>Component List</em> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

#### Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existante entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
la vue **Basic settings** du composant de connexion créant cette connexion.

2. **Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.**

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| **Username et Password** | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
**Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• **View schema** : sélectionnez cette option afin de voir le schéma.  
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
Table Name | Nom de la table à traiter.
---|---
Query type | Peut être **Built-in** ou **Repository**.

**Built-in** : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.

**Repository** : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.

Query | Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.

Die on error | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

**Propagate QUERY’s recordset** | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.

**Use PreparedStatement** | Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des ‘?’ dans l’instruction SQL définie dans le champ **Query** de l’onglet **Basic settings**.

**Parameter Index** : Saisissez la position du paramètre dans l’instruction SQL.

**Parameter Type** : Saisissez le type du paramètre.

**Parameter Value** : Saisissez la valeur du paramètre.

![Remarque](1)

Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

**Commit every** | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

**tStatCatcher Statistics** | Cochez cette case pour collector les données de log au niveau du composant.
Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
</tbody>
</table>

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
</table>

Scénarios associés

Pour un scénario associé, consultez :

- Procédure du composant tDBSQLRow.
- Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du composant tMysqlRow.
tIngresSCD

Reflette et traque les modifications d’une table Ingres SCD dédiée.

tIngresSCD répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.

Propriétés du tIngresSCD Standard

Ces propriétés sont utilisées pour configurer le tIngresSCD s’exécutant dans le framework de Jobs Standard.

Le composant tIngresSCD Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<p>| Property type | Peut être Built-in ou Repository. |</p>
<table>
<thead>
<tr>
<th><strong>Built-in</strong></th>
<th>Propriétés utilisées ponctuellement.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Server</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| **Username et Password** | Informations d’authentification de l’utilisateur de la base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Table** | Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. |
| **Built-in** | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| **Repository** | Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |
| **SCD Editor** | L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. |
Pour plus d’informations, consultez Méthodologie de gestion du SCD à la page 2716.

<table>
<thead>
<tr>
<th>Use memory saving Mode</th>
<th>Cochez cette case pour améliorer les performances du système.</th>
</tr>
</thead>
</table>
| Source keys include Null | Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null.  
**Avertissement :**  
Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique. |
| Die on error | Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. |

**Advanced settings**

| End date time details | Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est 12:00:00.  
Ce champ apparaît uniquement lorsqu’un SCD de *Type 2* est utilisé et lorsque *Fixed year value* est sélectionné pour créer la date de fin du SCD. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug mode</td>
<td>Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Variables globales**

| Global Variables |  
--- | ---  
**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable *After* et retourne un entier.  
**NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable *After* et retourne un entier.  
**NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable *After* et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  
Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type <strong>Row Main</strong> en entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez tMysqlSCD à la page 2712.
tInterbaseClose

Ferme la connexion à une base de données Interbase connectée.

Propriétés du tInterbaseClose Standard

Ces propriétés sont utilisées pour configurer le tInterbaseClose s’exécutant dans le framework de Jobs Standard.

Le composant tInterbaseClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tInterbaseConnection dans la liste.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
Lorsqu'un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tInterbaseCommit**

Commite en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes, permettant un gain de performance.

tInterbaseCommit valide les données traitées dans un Job à partir d’une base de données connectée.

**Propriétés du tInterbaseCommit Standard**

Ces propriétés sont utilisées pour configurer le tInterbaseCommit s’exécutant dans le framework de Jobs Standard.

Le composant tInterbaseCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td><strong>Component list</strong></td>
<td>Sélectionnez le composant <strong>tInterbaseConnection</strong> dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td><strong>Close Connection</strong></td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. <strong>Avertissement :</strong> Si vous utilisez un lien de type <strong>Row &gt; Main</strong> pour relier le <strong>tInterbaseCommit</strong> à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case <strong>Close connection</strong> car la connexion sera fermée avant la fin du commit de votre première ligne.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants Interbase et notamment avec les composants <code>tInterbaseConnection</code> et <code>tInterbaseRollback</code>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton <code>[+]</code> pour ajouter une ligne à la table. Dans le champ <code>Code</code>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste <code>Component List</code> de la vue <code>Basic settings</code> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Scénario associé

Pour un scénario associé au `tInterbaseCommit`, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tInterbaseConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

tInterbaseConnection ouvre une connexion vers une base de données afin d’effectuer une transaction.

**Propriétés du tInterbaseConnection Standard**

Ces propriétés sont utilisées pour configurer le tInterbaseConnection s’exécutant dans le framework de Jobs Standard.

Le composant tInterbaseConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Table Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
Use or register a shared DB Connection

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ Shared DB Connection Name qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l'exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options Use dynamic job et Use an independent process to run subjob du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.

Advanced settings

Auto Commit

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commute pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

Cochez cette case pour collecter les données de log au niveau du composant.

Utilisation

Règle d’utilisation

Ce composant est généralement utilisé avec des composants Interbase, notamment les composants tInterbaseCommit et tInterbaseRollback.

Limitation

Ce composant requiert l’installation des fichiers .jar liés.

Scénarios associés

Pour un scénario associé au composant tInterbaseConnection, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tInterbaseInput

Ce composant lit une base de données Interbase et en extrait des champs à l’aide de requêtes.
Le tInterbaseInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

Propriétés du tInterbaseInput Standard

Ces propriétés sont utilisées pour configurer le tInterbaseInput s’exécutant dans le framework de Jobs Standard.
Le composant tInterbaseInput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>![Assistant]</td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td></td>
<td>Remarque : Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une</td>
</tr>
</tbody>
</table>
connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

---

**Host**
Adresse IP du serveur de base de données.

**Database**
Nom de la base de données.

**Username et Password**
Informations d’authentification de l’utilisateur de base de données.
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Schema et Edit Schema**
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

- **Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.
- **Repository** : Le schéma existe déjà et est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job...
Courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Query type et Query**

Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.

**Advanced settings**

<table>
<thead>
<tr>
<th>Paramètre</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim all the String/Char columns</td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td>Trim column</td>
<td>Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td>QUERY</td>
<td>Requête traitée. Cette variable est une variable <strong>Flow</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option. Une variable <strong>Flow</strong> fonctionne durant l'exécution d'un composant. Une variable <strong>After</strong> fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Interbase.</td>
<td></td>
</tr>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant...</td>
</tr>
</tbody>
</table>
la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Limitation

Ce composant requiert l’installation des fichiers .jar liés.

### Scénarios associés

Pour consulter les scénarios associés :

Consultez également le scénario du composant **tContextLoad** : **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520.
tInterbaseOutput

Exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

*tlInterbaseOutput* écrit, met à jour, modifie ou supprime les données d’une base de données.

**Propriétés du tlInterbaseOutput Standard**

Ces propriétés sont utilisées pour configurer le tlInterbaseOutput s’exécutant dans le framework de Jobs Standard.

Le composant *tlInterbaseOutput Standard* appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez [Composants de bases de données dynamiques](#) à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><img src="image" alt="Assistant de configuration de connexion" /></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue <strong>Basic settings</strong> du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><img src="image" alt="Remarque" /></td>
<td><strong>Remarque :</strong> Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par...</td>
</tr>
</tbody>
</table>
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td><strong>None</strong> : n’effectuer aucune opération de table.</td>
</tr>
<tr>
<td></td>
<td><strong>Drop and create the table</strong> : supprimer la table puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td><strong>Create a table</strong> : créer une table qui n’existe pas encore.</td>
</tr>
<tr>
<td></td>
<td><strong>Create table if doesn’t exist</strong> : créer la table si nécessaire.</td>
</tr>
<tr>
<td></td>
<td><strong>Drop a table if exists and create</strong> : supprimer la table si elle existe déjà, puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td><strong>Clear a table</strong> : supprimer le contenu de la table.</td>
</tr>
<tr>
<td>Action on data</td>
<td>Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td><strong>Insert</strong> : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.</td>
</tr>
<tr>
<td></td>
<td><strong>Update</strong> : Mettre à jour les entrées existantes.</td>
</tr>
<tr>
<td></td>
<td><strong>Insert or update</strong> : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.</td>
</tr>
<tr>
<td></td>
<td><strong>Update or insert</strong> : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.</td>
</tr>
</tbody>
</table>
**Delete** : Supprimer les entrées correspondantes au flux d’entrée.

**Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton \[...\] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

<table>
<thead>
<tr>
<th>Clear data in table</th>
<th>Cochez cette case pour que les données soient supprimées de la table sélectionnée avant toute action.</th>
</tr>
</thead>
</table>

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

**Commit every**

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

**Additional Columns**

Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétreatment particulier.

**Name** : Saisissez le nom de la colonne à modifier ou à insérer.

**SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.

**Position** : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l’action à effectuer sur la colonne de référence.

**Reference column** : Saisissez une colonne de référence que le composant **tInterbaseOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

**Use field options**

Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.

**Enable debug mode**

Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.

**Use Batch**

Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.

**Remarque** :
Cette case est disponible lorsque vous sélectionnez Insert, Update, ou Delete dans la liste Action on data.

<table>
<thead>
<tr>
<th>Batch Size</th>
<th>Spécifiez le nombre d'enregistrements à traiter dans chaque lot. Ce champ est disponible uniquement lorsque la case Use batch mode est cochée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable Flow fonctionne durant l'exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l'aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données Interbase. Il permet aussi de créer un flux de rejet avec un lien Row &gt; Reject filtrant les données en erreur. Pour un exemple d'utilisation, consultez Scénario : Récupérer</th>
</tr>
</thead>
</table>


### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


### Limitation

Ce composant requiert l’installation des fichiers .jar liés.

---

**Scénarios associés**

Pour un scénario associé, consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667 associé au composant tMysqlOutput.
tInterbaseRollback

Evite le commit de transaction involontaire en annulant la transaction dans une base de données Interbase connectée.

Propriétés du tInterbaseRollback Standard

Ces propriétés sont utilisées pour configurer le tInterbaseRollback s’exécutant dans le framework de Jobs Standard.

Le composant tInterbaseRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th><strong>Database</strong></th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Component list</strong></td>
<td>Sélectionnez le composant de connexion tInterbaseConnection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td><strong>Close Connection</strong></td>
<td>Décrochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th><strong>tStatCatcher Statistics</strong></th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

Utilisation

<table>
<thead>
<tr>
<th><strong>Règle d’utilisation</strong></th>
<th>Ce composant est généralement utilisé en association avec des composants Interbase, notamment avec le tInterbaseConnection et le tInterbaseCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez</td>
</tr>
</tbody>
</table>
Scénario associé

Pour un scénario associé au tInterbaseRollback, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623 du composant tMySQLRollback.
**tlInterbaseRow**

Agit sur la structure même de la base de données ou sur les données (mais sans les manipuler) en utilisant SQLBuilder écrire rapidement et aisément vos requêtes.

tlInterbaseRow exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

**Propriétés du tlInterbaseRow Standard**

Ces propriétés sont utilisées pour configurer le tlInterbaseRow s'exécutant dans le framework de Jobs Standard.

Le composant tlInterbaseRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

<i>Remarque :</i>

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

<i>Remarque :</i>

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
la vue **Basic settings** du composant de connexion créant cette connexion.

2. **Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.**

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table Name</td>
<td>Nom de la table à traiter.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Le schéma est créé et conservé pour ce composant seulement. Voir également le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <strong>No</strong> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
</tbody>
</table>
### Query type
La requête peut être **Built-in** ou distante dans le **Repository**

**Built-in** : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.

**Repository** : Sélectionnez la requête appropriée dans le **Repository**. Le champ **Query** est renseigné automatiquement.

### Query
Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.

### Die on error
Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

<table>
<thead>
<tr>
<th>Propagate QUERY’s recordset</th>
<th>Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste <strong>use column</strong>.</th>
</tr>
</thead>
</table>
|                             | **Remarque** :
|                             | Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type **Object**. Ce composant est généralement suivi du tParseRecordSet. |

<table>
<thead>
<tr>
<th>Use PreparedStatement</th>
<th>Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau <strong>Set PreparedStatement Parameter</strong>, définissez les valeurs des paramètres représentés par des “?” dans l’instruction SQL définie dans le champ <strong>Query</strong> de l’onglet <strong>Basic settings</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Parameter Index</strong> : Saisissez la position du paramètre dans l'instruction SQL.</td>
</tr>
<tr>
<td></td>
<td><strong>Parameter Type</strong> : Saisissez le type du paramètre.</td>
</tr>
<tr>
<td></td>
<td><strong>Parameter Value</strong> : Saisissez la valeur du paramètre.</td>
</tr>
</tbody>
</table>
|                         | **Remarque** :
|                         | Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance. |

| Commit every | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution. |
### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable.</td>
</tr>
<tr>
<td></td>
<td>Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez <strong>Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte</strong> à la page 2641 et <strong>Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement</strong> à la page 520. Pour plus d’informations concernant les...</td>
</tr>
</tbody>
</table>
Scénarios associés

Pour un scénario associé, consultez :

- **Scénario : Combiner deux flux pour une sortie sélective** à la page 2706.
- **Procédure associé au composant** tDBSQLRow.
- **Scénario : Supprimer et re-générer un index de table MySQL** à la page 2700 associé au composant tMysqlRow.
**tIntervalMatch**

Retourne une valeur à l'aide d'une jointure.

tIntervalMatch permet d’établir une jointure entre un flux principal et un flux de référence. Il classe une valeur spécifiée du flux principal dans la fourchette de valeurs correspondante définie dans le flux de référence et retourne les informations correspondantes.

**Propriétés du tIntervalMatch Standard**

Ces propriétés sont utilisées pour configurer le tIntervalMatch s’exécutant dans le framework de Jobs Standard.

Le composant tIntervalMatch Standard appartient à la famille Data Quality.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit schema                                                                CNT-6552     |
|---------------------------------------------------------------------------------------|-------------|
| Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs. |


<table>
<thead>
<tr>
<th>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</th>
</tr>
</thead>
<tbody>
<tr>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Search Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez la colonne du flux principal contenant les valeurs qui doivent être mises en correspondance avec l’intervalle de valeurs.</td>
</tr>
</tbody>
</table>
**Column (LOOKUP)**
Sélectionnez la colonne du flux lookup contenant les valeurs qui doivent être retournées quand la jointure est ok.

**Lookup Column (min) / Include the bound (min)**
Sélectionnez la colonne contenant la valeur minimale de l'intervalle. Cochez la case pour inclure la valeur maximale de l'intervalle.

**Lookup Column (max) / Include the bound (max)**
Sélectionnez la colonne contenant la valeur maximale de l'intervalle. Cochez la case pour inclure la valeur maximale de l'intervalle.

**Advanced settings**

**tStatCatcher Statistics**
Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Global Variables**

**Global Variables**

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

**Règle d’utilisation**
Ce composant manipule un flux de données, il requiert donc un Input et un Output, et nécessite la définition d’une étape intermédiaire.

**Scénario : Identifier l’emplacement de serveurs à partir de leur IP**

Ce scénario décrit un Job à quatre composants vérifiant les adresses IP de serveur, listées dans le fichier d’entrée principal, par rapport à une liste d’intervalles d’IPs dans le fichier de référence, afin d’identifier le pays d’hébergement de chaque serveur.
**Construire le Job**

**Procédure**

1. Dans l'espace de modélisation, glissez deux composants `tFileInputDelimited`, un `tIntervalMatch` et un `tLogRow`.
2. Reliez les composants à l'aide de liens `Row > Main`.
   - La connexion du second `tFileInputDelimited` au `tIntervalMatch` apparaît comme `Lookup`.

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le premier `tFileInputDelimited` pour ouvrir sa vue `Component` et configurer ses propriétés.

   ![Component Configuration](image)

   - Parcourez votre système jusqu'au fichier d'entrée principal, contenant une liste des serveurs et de leur adresse IP:
     ```
     Server1;057.010.010.010
     Server2;001.010.010.100
     Server3;057.030.030.030
     Server4;053.010.010.100
     ```

   2. Cliquez sur le bouton ` [...]` à côté du champ `Edit schema` pour ouvrir la boîte de dialogue `[Schema]` et configurer le schéma d'entrée. Selon la structure du fichier d'entrée, le schéma se compose...
de deux colonnes, Server et IP, de type \textit{String}. Cliquez sur \textbf{OK} pour fermer la boîte de dialogue.

4. Dans le champ \textit{Header}, saisissez le nombre de lignes d’en-tête à ignorer et laissez les autres champs tels qu’ils sont.

5. Configurez de la même manière les propriétés du second \texttt{tFileInputDelimited}.

Le fichier à utiliser en entrée pour le flux de référence liste, dans cet exemple, des intervalles d’adresses IP et le pays correspondant.

\begin{verbatim}
StartIP;EndIP;Country
001.000.000.000;001.255.255.255;USA
002.006.190.056;002.006.190.063;UK
011.000.000.000;011.255.255.255;USA
057.000.000.000;057.255.255.255;France
012.063.178.060;012.063.178.063;Canada
053.000.000.000;053.255.255.255;Germany
\end{verbatim}

Le schéma du flux de référence doit posséder la structure suivante
6. Double-cliquez sur le composant **tIntervalMatch** pour ouvrir sa vue **Basic settings**.


8. Dans la liste **Column (LOOKUP)**, sélectionnez la colonne de référence contenant les valeurs à retourner. Dans cet exemple, il s’agit du nom des pays où se trouvent les serveurs.

9. Configurez les colonnes de référence minimale et maximale correspondant aux limites de l’intervalle défini dans le schéma de référence, **StartIP** et **EndIP**, respectivement.

**Exécuter le Job**

**Procédure**

Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job et appuyez sur **F6** pour l’exécuter.

Le nom du pays où est hébergé chaque serveur s’affiche à côté de l’adresse IP correspondante, dans la console.
**tliterateToFlow**

Transforme des données non traitables en flux traitable.

tliterateToFlow transforme une liste en flux de données pouvant être traité.

**Propriétés du tliterateToFlow Standard**

Ces propriétés sont utilisées pour configurer le tliterateToFlow s'exécutant dans le framework de Jobs Standard.

Le composant tliterateToFlow Standard appartient à la famille Orchestration.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma est créé et conservé pour ce composant uniquement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
| Mapping | **Column** : Saisissez le nom des colonnes à créer.
**Value** : Appuyez sur Ctrl+Espace pour accéder à toutes les variables disponibles soit globales soit celles que vous avez définies. |
Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant n’est pas un composant de début (fond vert) et requiert un composant de sortie. |

Scénario : Transformer une liste de fichiers en flux de données

Le scénario suivant décrit un Job permettant de faire une boucle sur une liste de fichiers, de récupérer leur nom, ainsi que la date en cours, pour ensuite transformer ces données en un flux qui sera affiché dans la console.

- Cliquez et déposez les composants suivants : tFileList, tIterateToFlow et tLogRow.
• Connectez le **tFileList** au **tIterateToFlow** à l’aide d’un lien **Iterate** et connectez le Job au composant **tLogRow** à l’aide d’un lien **Row Main**.

• Dans la vue **Component** du composant **tFileList**, donnez le chemin d’accès au répertoire contenant la liste de fichiers.

![Component tFileList](image)

**Basic settings**
- **Directory**: “D:/Input/Countries”
- **FileList Type**: Files
- **Includes subdirectories**: Non
- **Case Sensitive**: No
- **Generate Error if no file found**: Non
- **Glob Expressions**: Use

**Advanced settings**
- **Files**:
  - **Filename**: Tous les fichiers dans le répertoire
  - **Date Format**: yy-dd-MM

• Dans cet exemple, les fichiers sont trois fichiers .txt simples stockés dans le répertoire **Countries**.

• Dans la liste **Case sensitive**, sélectionnez **No** (Non) pour ne pas tenir compte de la casse.

• Sélectionnez le composant **tIterateToFlow** et cliquez sur **Edit Schema** pour modifier le nouveau schéma.

**tIterateToFlow**

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Type</th>
<th>N.</th>
<th>Date Pattern (yyyy-MM-dd)</th>
<th>Length</th>
<th>Precision</th>
<th>Default</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filename</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td></td>
<td>Date</td>
<td></td>
<td>'dd-MM-yyyy'</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Ajoutez deux nouvelles colonnes : **Filename** de type **String** et **Date** de type **date**. Assurez-vous d’avoir paramétré le bon modèle en Java.

• Cliquez sur **OK** pour valider.

• Une boîte de dialogue s’ouvre et vous demande si vous souhaitez propager les changements. Cliquez sur **Yes** (Oui).

• Notez que ce nouveau schéma apparaît dans le tableau **Mapping**.

**Mapping**

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filename</td>
<td>(String)globalMap.get(&quot;tFileList_1_CURRENT_FILEPATH&quot;)</td>
</tr>
<tr>
<td>Date</td>
<td>TalendDate.getDate(&quot;CCYY-MM-DD hh:mm:ss&quot;)</td>
</tr>
</tbody>
</table>

• Dans chaque cellule de la colonne **Value**, appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales et à celles que vous avez définies.

• Pour la colonne **Filename**, utilisez la variable globale **tFileList_1\_CURRENT\_FILEPATH**. Elle reprend le chemin d’accès courant pour recueillir le nom de chaque fichier, le Job fait une boucle sur ce répertoire.

• Pour la colonne **Date**, utilisez la routine Talend : **TalendDate.getCurrentDate()**.
Dans la vue **Component** du composant **tLogRow**, cochez la case **Print values in cells of a table**.

Enregistrez votre Job et exécutez-le, en appuyant sur **F6**.

---

<table>
<thead>
<tr>
<th>tLogRow_1</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>D:\Input\Countries\in-01.txt</td>
<td>2010-03-10 10:28:56</td>
</tr>
<tr>
<td>D:\Input\Countries\in-02.txt</td>
<td>2010-03-10 10:28:56</td>
</tr>
<tr>
<td>D:\Input\Countries\in-03.txt</td>
<td>2010-03-10 10:28:56</td>
</tr>
</tbody>
</table>

Le chemin d'accès apparaît dans la colonne **Filename** et la date en cours apparaît dans la colonne **Date**.
**tJasperOutput**

Crée un rapport dans des formats divers en utilisant l’outil iReport de Jaspersoft.


tJasperOutput lit et traite des données d’un flux d’entrée afin de créer un rapport à l’aide d’un modèle de rapport JRXML défini dans iReport.

**Propriétés du tJasperOutput Standard**

Ces propriétés sont utilisées pour configurer le tJasperOutput s’exécutant dans le framework de Jobs Standard.

Le composant tJasperOutput Standard appartient à la famille Business Intelligence.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Temp path</td>
<td>Chemin d’accès aux fichiers temporaires.</td>
</tr>
<tr>
<td>Destination path</td>
<td>Chemin d’accès au fichier final du rapport.</td>
</tr>
<tr>
<td>File name/Stream</td>
<td>Nom du rapport final.</td>
</tr>
<tr>
<td>Report type</td>
<td>Type du fichier du rapport final.</td>
</tr>
</tbody>
</table>

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réserve line lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
| **Built-in** | Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| **Repository** | Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |

**Sync columns**
Permet de synchroniser le schéma du fichier d’entrée avec le schéma du fichier de sortie. La fonction Sync ne s’affiche que lorsqu’une connexion de type Row est connectée au composant de sortie.

**iReport**
Editez la commande pour fournir le chemin d'accès au fichier d'exécution de iReport, par exemple en remplaçant \_IREPORT\_PATH\_\_ par E:\Program Files \Vasipersoft\Report-4.1.1\bin\, ou en saisissant le chemin d'accès complet au fichier d'exécution, par exemple "E:\Program Files\Vasipersoft\Report-4.1.1\bin\\Report.exe".

**Launch**
Cliquez sur ce bouton pour lancer iReport.

### Advanced settings

**Specify Locale**
Cochez cette case afin de sélectionner la langue de votre rapport dans la liste Report Locale.

**Remarque**:

**Encoding**
Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace.
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Règle d’utilisation**

Ce composant est étroitement lié à l'outil de reporting de Jaspersoft, iReport. Il lit et traite des données d’un flux d’entrée afin de créer un rapport à l’aide d’un modèle de rapport JRXML défini dans iReport.

**Scénario : Générer un rapport avec un modèle .jrxml**

Le Job suivant lit des données d'un fichier .csv et crée un rapport .pdf basé sur un modèle de rapport .jrxml existant. Notez que le fichier de modèle doit avoir été créé via l'outil iReport de Jaspersoft à partir d'un fichier partageant le même schéma avec le fichier source .csv du Job.

**Configurer votre Job**

**Procédure**

2. Reliez le tFileInputDelimited au tJasperOutput à l'aide d'un lien Row.

**Configurer le composant d'entrée**

**Procédure**

1. Double-cliquez sur le composant tFileInputDelimited afin d'afficher sa vue Basic settings.
2. Sélectionnez Built-In dans la liste déroulante Property Type.

   Remarque :
   Vous pouvez sélectionner Repository dans la liste Property Type afin de renseigner automatiquement les champs correspondants si la métadonnée a été stockée localement dans le Repository. Pour plus d’informations concernant les métadonnées Metadata, consultez le Guide utilisateur du Studio Talend.

3. Renseignez le champ File name/Stream en saisissant le chemin d’accès et le nom du fichier source, par exemple "C:/Documents and Settings/Andy ZHANG/nom.csv".

4. Laissez les paramètres par défaut pour les champs Row Separator et Field Separator. Cependant, si nécessaire, vous pouvez les modifier selon vos besoins.

5. Saisissez 1 dans le champ Header et 0 dans le champ Footer. Laissez le champ Limit vide. Cependant, si nécessaire, vous pouvez les modifier selon vos besoins.


![Schema of FileInputDelimited_1](image)

**Configurer le composant de sortie**

**Procédure**

1. Double-cliquez sur le tJasperOutput pour afficher sa vue Basic settings et configurer ses propriétés.

![tJasperOutput](image)


   Remarque :
Le schéma du fichier utilisé pour créer un modèle .jrxml via iReport, doit être le même que celui du fichier source utilisé pour créer le rapport.


4. Saisissez le chemin d'accès au rapport final généré durant l'exécution du Job, dans le champ **Destination path**. Vous pouvez également cliquer sur le bouton [...] afin de parcourir votre système.

5. Saisissez le nom du rapport final généré durant l'exécution du Job, dans le champ **File name/Stream**.


7. Cliquez sur **Sync columns** afin de récupérer le schéma du composant précédent.

8. Saisissez le chemin d'accès au fichier d'exécution de iReport dans le champ **iReport**, en remplaçant par exemple \_IREPORT\_PATH\_ par E:\Program Files\jaspersoft\iReport-4.1.1\bin\, Vous pouvez cliquer sur le bouton **Launch** afin de lancer iReport.

   **Remarque :**
   Cette étape n’est pas obligatoire. Cependant, elle vous permet d’accéder à l’outil iReport pour effectuer les opérations nécessaires, comme la création d’un modèle de rapport, etc.

**Exécuter votre Job**

**Procédure**

1. Sauvegardez le Job.

2. Appuyez sur **F6** pour l'exécuter.
   Le fichier *out.pdf* est créé à l’emplacement spécifié dans le champ **Destination path**.
**tJasperOutputExec**

Crée un rapport dans des formats divers en utilisant l’outil iReport de Jaspersoft et permet un gain de performance, puisqu’il est la combinaison d’un composant d’entrée et du composant de sortie tJasperOutput.


Le composant tJasperOutputExec lit et traite des données à partir d’un fichier source pour créer un rapport à l’aide d’un modèle de rapport .jrxml défini via iReport.

**Propriétés du tJasperOutputExec Standard**

Ces propriétés sont utilisées pour configurer le tJasperOutputExec s’exécutant dans le framework de Jobs Standard.

Le composant tJasperOutputExec Standard appartient à la famille Business Intelligence. Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jrxml file</td>
<td>Fichier du modèle de rapport créé dans iReport</td>
</tr>
<tr>
<td>Source file</td>
<td>Nom du fichier source</td>
</tr>
<tr>
<td>Record delimiter</td>
<td>Saisissez votre séparateur d’enregistrements.</td>
</tr>
<tr>
<td>Destination path</td>
<td>Chemin d’accès au fichier final du rapport.</td>
</tr>
<tr>
<td>Use Default Output Name</td>
<td>Cochez cette case pour utiliser le nom par défaut du rapport généré, c’est-à-dire le nom du fichier source.</td>
</tr>
<tr>
<td>Output Name</td>
<td>Nom du rapport final</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Ce champ n’apparaît pas si la case <strong>Use Default Output Name</strong> est cochée.</td>
</tr>
<tr>
<td>Report type</td>
<td>Type du fichier du rapport final.</td>
</tr>
<tr>
<td>iReport</td>
<td>Editez la commande pour fournir le chemin d’accès au fichier d’exécution de iReport, par exemple en remplaçant _IREPORT_PATH_ par E:\Program Files Jaspersoft\Report-4.1.1\bin, ou en saisissant le chemin d’accès complet au fichier d’exécution, par exemple “E:\Program Files\Jaspersoft\Report-4.1.1\bin\Report.exe”.</td>
</tr>
<tr>
<td>Launch</td>
<td>Cliquez sur ce bouton pour lancer iReport.</td>
</tr>
</tbody>
</table>
### Advanced settings

| **Specify Locale** | Cochez cette case afin de sélectionner la langue de votre rapport dans la liste *Report Locale*.  
**Remarque:**  
La première ligne de la liste *Report Locale* est vide. Vous pouvez cliquer dessus pour personnaliser le langage. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Encoding</strong></td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez <em>Custom</em> et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

| **Global Variables** | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  
Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |
|-------------------|--------------------------------------------------------------------------------------------------|

### Utilisation

<table>
<thead>
<tr>
<th><strong>Règle d’utilisation</strong></th>
<th>Ce composant est étroitement lié à l’outil de reporting de Jaspersoft, iReport. Il lit et traite des données à partir d’un fichier source pour créer un rapport à l’aide d’un modèle de rapport .jrxml défini via iReport.</th>
</tr>
</thead>
</table>

### Scénario associé

Pour un scénario associé, consultez *Scénario : Générer un rapport avec un modèle .jrxml* à la page 1934.
tJava

Etend les fonctionnalités du Job Talend grâce au langage Java.

tJava permet de saisir du code personnalisé afin de l'intégrer dans le programme Talend. Ce code est exécuté une seule fois.

Propriétés du tJava Standard

Ces propriétés sont utilisées pour configurer le tJava s'exécutant dans le framework de Jobs Standard. Le composant tJava Standard appartient à la famille Custom Code. Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisissez le code Java que vous souhaitez exécuter. Pour plus d'informations concernant la syntaxe des fonctions Java spécifiques à Talend, consultez l'aide en ligne du Studio Talend dans le menu Help Contents &gt; Developer Guide &gt; API Reference. Pour la référence Java complète, consultez <a href="http://docs.oracle.com/javaee/6/api/">http://docs.oracle.com/javaee/6/api/</a> (en anglais).</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Import</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisissez le code Java permettant d’importer, si nécessaire, la librairie externe utilisée dans le champ Main code de l’onglet Basic settings.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
</tr>
</thead>
</table>
Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est généralement utilisé en tant que sous-job à un seul composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Il est nécessaire de connaître le langage Java.</td>
</tr>
</tbody>
</table>

Scénario : Imprimer le contenu d'une variable

Le scénario suivant est une simple démonstration de l'application possible du composant **tJava**. L'objectif de ce Job est d'imprimer les lignes traitées à l'aide d'une commande Java et des variables globales fournies dans le *Studio Talend*.

Construire le Job

Procédure

1. Cliquez et déposez les composants dans l'espace de modélisation : **tFileInputDelimited**, **tFileOutputExcel**, **tJava**.
2. Connectez le composant **tFileInputDelimited** au **tFileOutputExcel** à l'aide d'une connexion de type *Row Main*. Le contenu du fichier .txt délimité sera transmis à un fichier de type XSL via cette connexion sans subir de transformation.
3. Ensuite, connectez le composant **tFileInputDelimited** au composant **tJava** à l'aide d'un lien de type *Trigger > On Subjob Ok*. Ce lien met en place un séquençage ordonnant au **tJava** de se lancer à la fin du processus principal.

Configurer le composant d’entrée

Procédure

1. Définissez les paramètres du composant **tFileInputDelimited** dans l'onglet *Basic settings* de la vue **Component**. Le fichier d'entrée utilisé en exemple est un simple fichier texte à deux colonnes : Name (des noms de personnes) et leur Emails respectifs.
2. Pour cet exemple, le schéma n'a pas été centralisé dans le référentiel, vous devez donc configurer manuellement le schéma de ces deux colonnes.

3. Cliquez sur le bouton **Edit Schema**.

4. Une boîte de dialogue vous demandant si vous souhaitez propager les modifications apportées apparaît, cliquez sur **OK** pour accepter la propagation. Ainsi, le schéma du composant **tFileOutputExcel** est automatiquement renseigné à partir du schéma du composant précédent. Vous n'aurez donc pas à définir le schéma de nouveau.

**Configurer le composant de sortie**

Paramétrez le fichier de sortie afin qu'il reçoive le contenu du fichier d'entrée sans modifications. Si le fichier n'existe pas déjà, il sera créé.
Dans cet exemple, dans le champ **Sheet name**, indiquez le nom de la feuille Excel, ici **Email** et la case **Header** est cochée pour indiquer que le fichier contient un en-tête.

## Configurer le composant tJava

### Procédure

1. Sélectionnez le composant **tJava** pour définir la commande Java à exécuter.

2. Dans la zone **Code**, saisissez la commande suivante :

   ```java
   String var = "Nb of line processed: ";
   var = var + globalMap.get("tFileInputDelimited_1_NB_LINE");
   System.out.println(var);
   ```

## Résultats

Dans cet exemple précis, utilisez la variable **NB_Line**. Pour accéder à la liste des variables globales, appuyez sur **Ctrl + Espace** de votre clavier et sélectionnez la variable globale correspondante.
Exécuter le Job

Procédure

1. Enregistrez le Job en appuyant sur Ctrl+S.

Résultats

Nb of line processed: 4
Job JavaDb ended at 13:53 20/08/2007. [exit code=0]

Le contenu est donc transmis au fichier Excel spécifié et le nombre de lignes traitées est affiché dans la console Run.
tJavaDBInput

Lit une base de données et en extrait des champs à l’aide de requêtes.

tJavaDBInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

Propriétés du tJavaDBInput Standard

Ces propriétés sont utilisées pour configurer le tJavaDBInput s'exécutant dans le framework de Jobs Standard.

Le composant tJavaDBInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Framework</td>
<td>Sélectionnez dans la liste un framework pour votre base de données Java.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>DB root path</td>
<td>Parcourez vos fichiers jusqu’à la racine de votre base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés</td>
</tr>
</tbody>
</table>
au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

**Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.


Cliquez sur `Edit schema` pour modifier le schéma. Si le schéma est en mode `Repository`, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode `Built-In` et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur `No` et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre `Repository Content`.

**Query type et Query**

Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

**Advanced settings**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trim all the String/Char columns</strong></td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td><strong>Trim column</strong></td>
<td>Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>Nombre de lignes traitées. Cette variable est une variable <code>After</code> et retourne un entier.</td>
</tr>
<tr>
<td><strong>QUERY</strong></td>
<td>Requête traitée. Cette variable est une variable <code>Flow</code> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <code>After</code> et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données JavaDB.</th>
</tr>
</thead>
</table>

| Limitation          | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center ([https://help.talend.com](https://help.talend.com)). |

### Scénarios associés

Pour des scénarios associés, consultez :

Consultez également le scénario associé du composant **tContextLoad** : **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520.
**tJavaDBOutput**

Exécute l’action défini sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

tJavaDBOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

**Propriétés du tJavaDBOutput Standard**

Ces propriétés sont utilisées pour configurer le tJavaDBOutput s’exécutant dans le framework de Jobs Standard.

Le composant tJavaDBOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><img src="image" alt="Icone" /></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Framework</td>
<td>Sélectionnez dans la liste un framework pour votre base de données Java.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>DB root path</td>
<td>Parcourez vos fichiers jusqu’à la racine de votre base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :
| | **None** : n'effectuer aucune opération de table.
| | **Drop and create the table** : supprimer la table puis en créer une nouvelle.
| | **Create a table** : créer une table qui n'existe pas encore.
| | **Create table if doesn’t exist** : créer la table si nécessaire.
| | **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.
| | **Clear a table** : supprimer le contenu de la table.

| Action on data | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :
| | **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.
| | **Update** : Mettre à jour les entrées existantes.
| | **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
| | **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
| | **Delete** : Supprimer les entrées correspondantes au flux d’entrée.
| | **Avertissement** :
| | Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
### Built-In
Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

### Repository
Le schéma existe déjà et il est stocké dans le *Repository*. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center ([https://help.talend.com](https://help.talend.com)).

Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre *Repository Content*.

### Die on error
Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien *Row > Rejects*.

---

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Commit every</strong></th>
<th>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Additional Columns</strong></td>
<td>Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</td>
</tr>
</tbody>
</table>
**Name**: Saisissez le nom de la colonne à modifier ou à insérer.

**SQL expression**: Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.

**Position**: Sélectionnez **Before**, **Replace** ou **After**, en fonction de l’action à effectuer sur la colonne de référence.

**Reference column**: Saisissez une colonne de référence que le composant **tJavaDBOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

**Use field options**

Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.

**Enable debug mode**

Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong> : nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong> : nombre de lignes mises à jour. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong> : nombre de lignes insérées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong> : nombre de lignes supprimées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_REJECTED</strong> : nombre de lignes rejetées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
<td></td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données JavaDB. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.

Limitation


Scénarios associés

Pour un scénario associé, consultez :

• Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667 du composant tMysqlOutput.
tJavaDBRow

Selon la nature de la requête et de la base de données, tJavaDBRow agit sur la structure même de la base de données ou sur les données (mais sans les manipuler) en utilisant le SQLBuilder pour écrire vos requêtes rapidement et aisément.

tJavaDBRow exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tJavaDBRow Standard

Ces propriétés sont utilisées pour configurer le tJavaDBRow s'exécutant dans le framework de Jobs Standard.

Le composant tJavaDBRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td>Framework Type</td>
<td>Sélectionnez dans la liste un framework pour votre base de données Java.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>DB root path</td>
<td>Parcourez vos fichiers jusqu’à la racine de votre base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
</tbody>
</table>
### Built-in
Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Nom de la table à traiter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query type</td>
<td>La requête peut être Built-in ou distante dans le Repository</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.</td>
</tr>
<tr>
<td>Query</td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décocochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Rejects.</td>
</tr>
</tbody>
</table>

### Advanced settings

<p>| Propagate QUERY’s recordset | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column. |
| Use PreparedStatement | Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de |</p>
<table>
<thead>
<tr>
<th>Commit every</th>
<th>Commit every est un nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton Install dans l’onglet Component. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet Modules de</td>
</tr>
</tbody>
</table>
Scénarios associés

Pour un scénario associé, consultez :

• **Procédure** du composant *tDBSQLRow*.

• **Scénario** : Supprimer et re-générer un index de table MySQL à la page 2700 du composant *tMysqlRow*.
tJavaFlex

Fournit un éditeur de code Java qui permet de saisir du code personnalisé afin de l’intégrer dans le programme Talend.

Le tJavaFlex permet d’ajouter du code dans ses sections de code Start/Main/End. Avec le tJavaFlex, vous pouvez saisir les trois différentes parties du code Java (start, main et end) constituant un composant qui permet d’exécuter une opération souhaitée.

Propriétés du tJavaFlex Standard

Ces propriétés sont utilisées pour configurer le tJavaFlex s’exécutant dans le framework de Jobs Standard.

Le composant tJavaFlex Standard appartient à la famille Custom Code.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Sync columns pour récupérer le schéma du composant précédent.</td>
<td></td>
</tr>
</tbody>
</table>

|----------|--------------------------------------------------------------------------------------------------|

|-----------|--------------------------------------------------------------------------------------------------|

<table>
<thead>
<tr>
<th>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</th>
</tr>
</thead>
<tbody>
<tr>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
</tbody>
</table>
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th>Data Auto Propagate</th>
<th>Cochez cette case pour propager automatiquement les données au composant suivant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start code</td>
<td>Saisissez le code Java qui va être appelé lors de la phase d'initialisation.</td>
</tr>
<tr>
<td>Main code</td>
<td>Saisissez le code Java qui va être appliqué pour chaque ligne du flux.</td>
</tr>
<tr>
<td>End code</td>
<td>Saisissez le code Java qui va être appelé lors de la phase de clôture.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Import</th>
<th>Saisissez le code Java permettant d’importer, si nécessaire, la librairie externe utilisée dans le champ Main code de l’onglet Basic settings.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

|------------------|----------------------------------------------------------------------------------------------------------------------------------|

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en tant que composant de début, de milieu et de fin de Job, ainsi qu’en tant que sous-job à un seul composant.</th>
</tr>
</thead>
</table>
Scénario 1 : Générer un flux de données

Ce scénario décrit un Job à deux composants dont le but est de générer un flux de trois lignes décrivant les différentes civilité : mademoiselle (Miss), madame (Mrs) et monsieur (Mr), et de les afficher dans la console.

Construire le Job

Procédure

1. Cliquez-glissez un composant **tJavaFlex** de la famille **Custom Code** et un composant **tLogRow** de la famille **Logs & Errors** dans le Job Designer.
2. Connectez-les via un lien **Row Main**.

Configurer le composant **tJavaFlex**

Procédure

1. Double-cliquez sur le composant **tJavaFlex** pour paramétrer ses propriétés.

2. Dans la liste **Schema**, sélectionnez le mode **Built-in** et cliquez sur le bouton [...] à côté du champ **Edit schema** pour décrire manuellement la structure des données à passer au composant suivant.
3. Cliquez sur le bouton [+ ] pour ajouter deux colonnes : key de type Integer et value de type String.

4. Cliquez OK pour valider les changements et fermer la boîte de dialogue.

5. Dans les propriétés du tJavaFlex, cochez la case Data Auto Propagate pour propager automatiquement les données au composant suivant.

Dans cet exemple, n’effectuez pas de transformation sur les données récupérées.

6. Dans le champ Start code, saisissez le code qui sera exécuté lors de la phase d’initialisation.

Ici, le code indique l’initialisation du composant tJavaFlex en affichant le message START et met en place la boucle et les variables utilisées par la suite dans le code Java :

```java
System.out.println("## START
#");
String [] valueArray = {"Miss", "Mrs", "Mr"};
for (int i=0;i<valueArray.length;i++) {
```

7. Dans le champ Main code, saisissez le code qui sera appliqué pour chaque ligne de données. Ici, l’objectif est de retourner chaque clé (key) avec sa valeur :

```java
row1.key = i;
row1.value = valueArray[i];
```

**Avertissement :**

Dans le champ Main code, "row1" correspond au nom du lien qui sort du tJavaFlex. Si vous renommez ce lien, vous devez modifier le code de ce champ.

8. Dans le champ End code, saisissez le code qui sera exécuté lors de la phase de clôture. Ici, l’accolade ferme la boucle et le code indique la fin de l’exécution du tJavaFlex en affichant le message END :

```java
}
```
9. Si nécessaire, assurez-vous que le schéma a bien été propagé au tLogRow en double-cliquant sur le composant et en cliquant sur le bouton [...] à côté de Edit schema.

Sauvegarder et exécuter le Job

Procédure

1. Enregistrez le Job en appuyant sur Ctrl+S.
2. Exécutez le Job en appuyant sur la touche F6 ou en cliquant sur le bouton Run, dans l’onglet Run.

La console affiche les trois civilités, ainsi que leur clés correspondantes.

Scénario 2 : Traiter des lignes de données avec le tJavaFlex

Ce scénario décrit un Job à deux composants générant des données de manière aléatoire, récupérant et traitant ces données ligne par ligne avec du code Java via le tJavaFlex.

Construire le Job

Procédure

1. Cliquez-glissez un composant tRowGenerator de la famille Misc et un composant tJavaFlex de la famille Custom Code dans le Job Designer.
2. Connectez-les via un lien Row > Main.
Configurer le composant d'entrée

**Procédure**

1. Double-cliquez sur le composant **tRowGenerator** pour ouvrir son éditeur.

   ![Schema](image1)

   2. Dans l'éditeur, cliquez sur le bouton [+ ] pour ajouter quatre colonnes : number, txt, date et flag.
3. Définissez le schéma en paramétrant ces quatre colonnes conformément à la capture d'écran précédente.

   4. Dans la colonne **Functions**, sélectionnez la fonction [...] dans la liste de chaque colonne.
5. Dans la colonne **Parameters**, saisissez 10 paramètres différents pour chaque colonne. Ces 10 paramètres correspondent aux données qui seront retournées de manière aléatoire lors de l'exécution du **tRowGenerator**.
6. Cliquez sur **OK** pour valider vos changements et fermer l’éditeur.

Configurer le composant **tJavaFlex**

**Procédure**

1. Double-cliquez sur le composant **tJavaFlex** pour paramétrer ses propriétés.

   ![tJavaFlex](image2)

   2. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
3. Dans le champ **Start code**, saisissez le code qui sera exécuté lors de la phase d’initialisation. Ici, le code indique l’initialisation du composant **tJavaFlex** en affichant le message START et définit une variable qui sera utilisée par la suite dans le code Java :

```java
System.out.println("## START\n");
int i = 0;
```

4. Dans le champ **Main code**, saisissez le code qui sera appliqué pour chaque ligne de données. Ici, l’objectif est d’indiquer le numéro de la ligne en partant de 0, puis afficher le numéro et le texte aléatoire que vous mettez en majuscules et la date aléatoire définis dans le **tRowGenerator**. Ensuite, créez une condition affichant si le statut est **true** (vrai) ou **false** (faux) et incrémentez le numéro de la ligne :

```java
System.out.print(" row" + i + ":");
System.out.print("# number:" + row1.number);
System.out.print("txt:" + row1.txt.toUpperCase());
System.out.print("date:" + row1.date);
if(row1.flag) System.out.println(" | flag: true");
else System.out.println(" | flag: false");
i++;
```

**Avertissement :**

Dans le champ **Main code**, "row1" correspond au nom du lien qui sort du **tJavaFlex**. Si vous renommez ce lien, il faut penser à modifier le code de ce champ.

5. Dans le champ **End code**, saisissez le code qui sera exécuté lors de la phase de clôture. Ici, le code indique la fin de l’exécution du composant **tJavaFlex** en affichant le message END :

```java
System.out.println("## END");
```
Sauvegarder et exécuter le Job

Procédure

1. Enregistrez votre Job en appuyant sur Ctrl+S.
2. Appuyez sur F6 ou cliquez sur Run dans l'onglet Run pour l'exécuter.

Starting job tJavaFlex_scenario2 at 18:35 02/09/2009.

## START
#
row0: # number: 10  | txt: TEXT5  | date: 2006-05-25  | flag: false
row1: # number: 7   | txt: TEXT7  | date: 2006-06-26  | flag: true
row2: # number: 5   | txt: TEXT2  | date: 2006-05-25  | flag: false
row3: # number: 6   | txt: TEXT1  | date: 2005-08-28  | flag: true
row4: # number: 8   | txt: TEXT9  | date: 2006-05-25  | flag: false
row5: # number: 9   | txt: TEXT1  | date: 2007-01-21  | flag: false
row6: # number: 7   | txt: TEXT5  | date: 2004-11-21  | flag: true
row7: # number: 9   | txt: TEXT4  | date: 2005-08-29  | flag: true
row8: # number: 4   | txt: TEXT6  | date: 2006-06-26  | flag: false
row9: # number: 3   | txt: TEXT10 | date: 2006-05-25  | flag: false
#
## END

Job tJavaFlex_scenario2 ended at 18:35 02/09/2009. [exit code=0]

Résultats

La console affiche les données générées aléatoirement et modifiées à l'aide du code Java dans le tJavaFlex.
tJavaRow

Fournit un éditeur de code qui vous permet de saisir le code Java qui va être appliqué pour chaque ligne du flux.

tJavaRow permet de saisir du code personnalisé afin de l’intégrer dans un programme Talend.

Propriétés du tJavaRow Standard

Ces propriétés sont utilisées pour configurer le tJavaRow s’exécutant dans le framework de Jobs Standard.

Le composant tJavaRow Standard appartient à la famille Custom Code.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets. Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com). |
|                      | Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
  • **View schema** : sélectionnez cette option afin de voir le schéma.  
  • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
  • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job |
courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent.

Generate code

Cliquez sur ce bouton afin de générer automatiquement le code dans le champ Code pour mapper les colonnes du schéma d’entrée à celles du schéma de sortie. Cette génération ne change rien à votre schéma.

Le principe de ce mapping est de relier les colonnes ayant le même nom. Vous pouvez adapter le code généra selon le mapping dont vous avez besoin.

Code

Saisissez le code Java qui va être appliqué pour chaque ligne du flux.

Advanced settings

Import

Saisissez le code Java permettant d’importer, si nécessaire, la librairie externe utilisée dans le champ Main code de l’onglet Basic settings.

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

Global Variables

NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaine de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Pour saisir une variable globale (par exemple COUNT dans le tFileRowCount) dans le champ Code, vous devez saisir manuellement le code complet, c'est-à-dire ((Integer)globalMap.get("tFileRowCount_COUNT")).
Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est utilisé en tant que composant de milieu de Job, et doit être relié à un composant de début et à un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Il est nécessaire de connaître le langage Java.</td>
</tr>
</tbody>
</table>

Scénario : Transformer des données ligne par ligne avec un tJavaRow

Dans ce scénario, les informations de quelques villes sont lues depuis un fichier d'entrée délimité et sont transformées à l'aide de code Java via le composant tJavaRow, puis affichées dans la console.

Construire le Job

Procédure

1. Déposez un composant tFileInputDelimited et un tJavaRow de la Palette dans l'espace de modélisation graphique et renommez-les afin d'identifier leur rôle dans le Job.
2. Connectez les deux composants à l'aide d'un lien Row > Main.

Configurer les composants

Procédure

1. Double-cliquez sur le tFileInputDelimited pour afficher sa vue Basic settings.

2. Dans le champ File name/Stream, saisissez le chemin d'accès au fichier d'entrée, entre guillemets doubles, ou parcourez votre système en cliquant sur le bouton [...] et saisissez 1 dans le champ Header.
Dans cet exemple, le fichier d’entrée contient :

```
City;Population;LandArea;PopDensity
Beijing;10233000;1418;7620
Moscow;10452000;1081;9644
Seoul;10422000;605;17215
Tokyo;8731000;617;14151
New York;8310000;789;10452
```


4. Double-cliquez sur le composant **tJavaRow** afin d’afficher sa vue **Basic settings**.

5. Cliquez sur **Sync columns** afin de vous assurer que le schéma est bien récupéré du composant précédent.

6. Dans le champ **Code**, saisissez le code à appliquer à chaque ligne de données.

Dans cet exemple, vous allez mettre des majuscules aux noms des villes, regrouper les nombres supérieurs à 1000 à l’aide de séparateurs de milliers pour une meilleure lecture et afficher les données dans la console :

```
System.out.print("\n + input_row.City.toUpperCase() + ":");
System.out.print(" - Population: " + FormatterUtils.format_Number(String.valueOf(input_row.Population), ',', ' people');
System.out.print(" - Land area: " + FormatterUtils.format_Number(String.valueOf(input_row.LandArea), ',', ' km2');
System.out.print(" - Population density: ");
```
Remarque :
Dans le champ Code, input_row fait référence à la connexion vers le tJavaRow.

Sauvegarder et exécuter le Job

Procédure
1. Appuyez sur les touches Ctrl+S afin de sauvegarder votre Job.
2. Appuyez sur F6 ou cliquez sur le bouton Run dans la vue Run pour exécuter le Job.

Les informations relatives aux villes sont transformées par le code Java défini dans le tJavaRow et affichées dans la console.
**tJDBCClose**

Ferme une connexion JDBC active afin de libérer des ressources occupées.

**Propriétés du tJDBCClose Standard**

Ces propriétés sont utilisées pour configurer le tJDBCClose s'exécutant dans le framework de Jobs Standard.

Le composant tJDBCClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Component</td>
<td>Sélectionnez dans la liste déroulante le composant établissant la connexion à fermer.</td>
</tr>
</tbody>
</table>

### Advanced settings

| tStatCatcher Statistics                      | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. |

### Variables globales

| ERROR_MESSAGE                                | Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez</td>
</tr>
</tbody>
</table>
accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.


Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tJDBCColumnList

Fait une liste des libellés des colonnes d'une table JDBC donnée.

tJDBCColumnList effectue une opération d’itération dans toutes les colonnes d’une table donnée, grâce à une connexion JDBC définie.

**Propriétés du tJDBCColumnList Standard**

Ces propriétés sont utilisées pour configurer le tJDBCColumnList s’exécutant dans le framework de Jobs Standard.

Le composant tJDBCColumnList Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Database Type</th>
<th>Sélectionnez le type de base de données à laquelle vous souhaitez vous connecter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tJDBCConnection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Table name</td>
<td>Nom de la table à lire.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.</td>
</tr>
<tr>
<td></td>
<td>Décocochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Reject.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>CURRENT_COLUMN : nom de la colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CURRENT_COLUMN_TYPE : ID du type de colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td>CURRENT_COLUMN_TYPE_NAME : nom du type de colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>
CURRENT_COLUMN_PRECISION : précision de la colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne un nombre entier.

CURRENT_COLUMN_SCALE : échelle de la colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne un nombre entier.

NB_COLUMN : nombre de colonnes itérées jusqu’à présent. Cette variable est une variable After et retourne un entier.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Il faut utiliser ce composant en association avec les autres composants JDBC, notamment avec le tJDBCConnection. |

**Scénario associé**

Pour un scénario associé, consultez Scénario : Itérer une table de base de données et lister le nom des colonnes de la table à la page 2611.
**tJDBCCommit**

Commite en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes, permettant ainsi un gain de performance.

tJDBCCommit valide les données traitées dans un Job à partir d’une base de données connectée.

### Propriétés du tJDBCCommit Standard

Ces propriétés sont utilisées pour configurer le tJDBCCommit s’exécutant dans le framework de Jobs Standard.

Le composant tJDBCCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

#### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Component</td>
<td>Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Cochez cette case pour fermer la connexion à la base de données une fois que le composant a terminé sa tâche. Décochez cette case pour continuer à utiliser la connexion sélectionnée, une fois que le composant a terminé sa tâche. Si ce composant est lié à votre Job par un lien <strong>Row &gt; Main</strong>, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case <strong>Close connection</strong> ou votre connexion sera fermée avant la fin du commit de la première ligne.</td>
</tr>
</tbody>
</table>

#### Advanced settings

| tStatCatcher Statistics   | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |
### Variables globales

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

### Utilisation

#### Règle d’utilisation

Ce composant est généralement utilisé avec des composants JDBC et notamment avec les composants tJDBCConnection et tJDBCRowback.

#### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.


### Scénario associé

Pour un scénario associé au composant tJDBCCommit, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tJDBCConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Propriétés du tJDBCConnection Standard

Ces propriétés sont utilisées pour configurer le tJDBCConnection s'exécutant dans le framework de Jobs Standard.

Le composant tJDBCConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• Built-In : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• Repository : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td>JDBC URL</td>
<td>URL JDBC de la base de données à utiliser. Par exemple, l'URL JDBC de la base de données Amazon Redshift est jdbc:redshift://endpoint:port/database.</td>
</tr>
<tr>
<td>Drivers</td>
<td>Renseignez cette table pour charger les pilotes Jar requis. Pour ce faire, cliquez autant de fois que nécessaire sur le bouton [+] sous la table, pour ajouter autant de lignes que vous le souhaitez, chaque ligne pour un pilote Jar. Sélectionnez ensuite la cellule et cliquez sur le bouton [...] à droite de la cellule, pour ouvrir l'assistant Module dans lequel vous pouvez sélectionner votre pilote Jar. Par exemple, le pilote Jar</td>
</tr>
<tr>
<td>RedshiftJDBC41-1.1.13.1013.jar pour la base de données Redshift.</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Driver Class</strong></td>
<td></td>
</tr>
<tr>
<td>Nom de la classe, entre guillemets doubles, pour le pilote spécifié. Par exemple, pour le pilote RedshiftJDBC41-1.1.13.1013.jar, le nom à saisir est com.amazon.redshift.jdbc41.Driver.</td>
<td></td>
</tr>
<tr>
<td><strong>Use Id et Password</strong></td>
<td></td>
</tr>
<tr>
<td>Données d’authentification de l’utilisateur à la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
<td></td>
</tr>
<tr>
<td><strong>Use or register a shared DB Connection</strong></td>
<td></td>
</tr>
<tr>
<td>Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ Shared DB Connection Name qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents. Cette option est incompatible avec les options <em>Use dynamic job</em> et <em>Use an independent process to run subjob</em> du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job. Cette case est indisponible lorsque la case <em>Specify a data source alias</em> est cochée.</td>
<td></td>
</tr>
<tr>
<td><strong>Specify a data source alias</strong></td>
<td></td>
</tr>
<tr>
<td>Cochez cette case et, dans le champ Data source alias qui s’affiche, spécifiez l’alias d’une source de données créée du côté Talend Runtime, pour utiliser le pool de connexions partagées défini dans la configuration de la source de données. Cette option fonctionne uniquement lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour un scénario associé, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647. Cette case est indisponible lorsque la case <em>Use or register a shared DB Connection</em> est cochée.</td>
<td></td>
</tr>
</tbody>
</table>

**Advanced settings**

| Use Auto-Commit |
| Cochez cette case pour activer le mode auto-commit. |
| **Auto Commit** |
| Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la... |
même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec des composants JDBC et notamment avec les composants tJDBCCommit et tJDBCRollback. |

### Scénario associé

Pour plus d’informations relatives au fonctionnement du composant tJDBCConnection, consultez tMysqlConnection à la page 2618.
tJDBCInput

Lit toute base de données via une API JDBC et en extrait des champs à l’aide de requêtes.

tJDBCInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

**Propriétés du tJDBCInput Standard**

Ces propriétés sont utilisées pour configurer le tJDBCInput s’exécutant dans le framework de Jobs Standard.

Le composant tJDBCInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td>• Built-In</td>
<td>les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>• Repository</td>
<td>les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste <strong>Connection Component</strong>.</td>
</tr>
<tr>
<td><img src="image" alt="Cliquez sur cette icône pour ouvrir l’assistant de connexion à la base de données et stocker les paramètres de connexion configurés dans la vue Basic settings." /></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de connexion à la base de données et stocker les paramètres de connexion configurés dans la vue <strong>Basic settings</strong>.</td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant la configuration et le stockage des paramètres de connexion à la base de données, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Connection Component</th>
<th>Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drivers</td>
<td>Renseignez cette table pour charger les pilotes Jar requis. Pour ce faire, cliquez autant de fois que nécessaire sur le bouton [+] sous la table, pour ajouter autant de lignes que vous le souhaitez, chaque ligne pour un pilote Jar. Sélectionnez ensuite la cellule et cliquez sur le bouton [...] à droite de la cellule, pour ouvrir l’assistant Module dans lequel vous pouvez sélectionner votre pilote Jar. Par exemple, le pilote Jar RedshiftJDBC41-1.1.13.1013.jar pour la base de données Redshift.</td>
</tr>
<tr>
<td>Driver Class</td>
<td>Nom de la classe, entre guillemets doubles, pour le pilote spécifié. Par exemple, pour le pilote RedshiftJDBC41-1.1.13.1013.jar, le nom à saisir est com.amazon.redshift.jdbc41.Driver.</td>
</tr>
<tr>
<td>Use Id et Password</td>
<td>Données d’authentification de l’utilisateur à la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| Schema et Edit schema| Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
  - **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
  Cliquez sur Edit schema pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.  
  - **View schema** : sélectionnez cette option afin de voir le schéma.  
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales. |
### Update repository connection

 Sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **Repository Content**.

### Table Name

Saisissez le nom de la table de laquelle récupérer les données.

### Query Type et Query

Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs correspond à celui défini dans le schéma.

- **Built-In** : saisissez la requête manuellement dans le champ **Query** ou cliquez sur le bouton [...] à côté du champ **Query** afin de construire l’instruction graphiquement, à l’aide du SQLBuilder.
- **Repository** : sélectionnez la requête correspondante stockée dans le référentiel en cliquant sur le bouton [...]. Dans la boîte de dialogue **Repository Content**, sélectionnez la requête à utiliser. Le champ **Query** est automatiquement renseigné.

### Guess Query

Cliquez sur ce bouton pour générer la requête dans le champ **Query** à partir de la table et du schéma définis.

### Guess Schema

Cliquez sur ce bouton pour générer les colonnes du schéma à partir de la requête définie dans le champ **Query**.

### Specify a data source alias

Cochez cette case et, dans le champ **Data source alias** qui s’affiche, spécifiez l’alias d’une source de données créée du côté Talend Runtime, pour utiliser le pool de connexions partagées défini dans la configuration de la source de données. Cette option fonctionne uniquement lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour un scénario associé, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647.

Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste déroulante **Connection Component**.

### Advanced settings

#### Use cursor

Cochez cette case pour spécifier le nombre de lignes avec lequel vous souhaitez travailler à tout moment. Cette option optimise les performances.

#### Trim all the String/Char columns

Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes de type String/Char.
### Check column to trim

Cochez cette case pour la colonne dont vous souhaitez supprimer les espaces en début et fin de champ.

Cette propriété n’est pas disponible lorsque la case Trim all the String/Char columns est cochée.

### Enable Mapping File for Dynamic

Cochez cette case pour utiliser le fichier de mappage de métadonnées spécifié lors de la lecture de données d’une colonne de type dynamique. Cette case est décochée par défaut.

Une fois cette case cochée, vous pouvez spécifier le fichier de mapping des métadonnées à utiliser, en sélectionnant un type de base de données dans la liste déroulante Mapping File.

Pour plus d’informations concernant les fichiers de mappage de métadonnées, consultez la section sur la conversion de type du Guide utilisateur du Studio Talend.

### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>QUERY</td>
<td>Instruction de requête en cours de traitement. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

**Règle d’utilisation**

Ce composant couvre toutes les possibilités de requête SQL dans les bases de données utilisant une connexion JDBC.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

**Scénarios associés**

Pour des scénarios associés, consultez :

Consultez également Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520.
tJDBCOutput

Exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Le composant tJDBCOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

Propriétés du tJDBCOutput Standard

Ces propriétés sont utilisées pour configurer le tJDBCOutput s’exécutant dans le framework de Jobs Standard.

Le composant tJDBCOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td>• Built-In</td>
<td>: les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>• Repository</td>
<td>: les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td>Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
<td></td>
</tr>
</tbody>
</table>

Cliquez sur cette icône pour ouvrir l’assistant de connexion à la base de données et stocker les paramètres de connexion configurés dans la vue Basic settings.
Pour plus d'informations concernant la configuration et le stockage des paramètres de connexion à la base de données, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th><strong>Connection Component</strong></th>
<th>Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JDBC URL</strong></td>
<td>URL JDBC de la base de données à utiliser. Par exemple, l'URL JDBC de la base de données Amazon Redshift est <code>jdbc:redshift://endpoint:port/database</code>.</td>
</tr>
<tr>
<td><strong>Drivers</strong></td>
<td>Renseignez cette table pour charger les pilotes Jar requis. Pour ce faire, cliquez autant de fois que nécessaire sur le bouton [+] sous la table, pour ajouter autant de lignes que vous le souhaitez, chaque ligne pour un pilote Jar. Sélectionnez ensuite la cellule et cliquez sur le bouton [...] à droite de la cellule, pour ouvrir l’assistant Module dans lequel vous pouvez sélectionner votre pilote Jar. Par exemple, le pilote Jar RedshiftJDBC41-1.1.13.1013.jar pour la base de données Redshift.</td>
</tr>
<tr>
<td><strong>Driver Class</strong></td>
<td>Nom de la classe, entre guillemets doubles, pour le pilote spécifié. Par exemple, pour le pilote RedshiftJDBC41-1.1.13.1013.jar, le nom à saisir est <code>com.amazon.redshift.jdbc41.Driver</code>.</td>
</tr>
<tr>
<td><strong>Use Id et Password</strong></td>
<td>Données d’authentification de l’utilisateur à la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table Name</strong></td>
<td>Saisissez le nom de la table dans laquelle écrire les données.</td>
</tr>
<tr>
<td><strong>Data Action</strong></td>
<td>Sélectionnez une action à effectuer sur les données de la table définie.</td>
</tr>
</tbody>
</table>

- **Insert** : ajoute de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
- **Update** : met à jour les entrées existantes.
- **Insert or update** : insère un nouvel enregistrement dans le pool d’index. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas dans le pool d’index, un nouvel enregistrement est inséré.
- **Delete** : supprime les entrées correspondantes au flux d’entrée.

**Avertissement :**
Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton `[...]` à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

<table>
<thead>
<tr>
<th><strong>Clear data in table</strong></th>
<th>Cochez cette case pour vider les données de la table avant d’effectuer l’action définie.</th>
</tr>
</thead>
</table>

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets. Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la
| **métadonnée du schéma dans la fenêtre [Repository Content].** |

| **Guess Schema** | Cliquez sur ce bouton pour générer les colonnes du schéma à partir des paramètres des colonnes de la table de base de données. |

| **Die on error** | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. 
Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. 
Lorsque des erreurs sont ignorées, vous pouvez collecter les lignes en erreur à l’aide d’un lien **Row > Reject**. |

| **Specify a data source alias** | Cochez cette case et, dans le champ **Data source alias** qui s’affiche, spécifiez l’alias d’une source de données créée du côté Talend Runtime, pour utiliser le pool de connexions partagées défini dans la configuration de la source de données. Cette option fonctionne uniquement lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour un scénario associé, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647. 
Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l’alias de la source de données spécifié. 
Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste déroulante **Connection Component**. |

---

**Advanced settings**

| **Commit every** | Spécifiez le nombre de lignes à traiter avant de commiter un lot de lignes dans la base de données. 
Cette option assure la qualité de la transaction (mais pas de rollback) et de meilleures performances lors de l’exécution. |

| **Additional Columns** | Cette option vous permet d’appeler des fonctions SQL afin d’effectuer des actions sur des colonnes, actions qui ne sont ni des insertions, ni des mises à jour, ni des suppressions, ni des actions demandant un pré-traitemet particulier. Cette option ne vous est pas proposée si vous créez (avec ou sans suppression) une table de base de données. 
• **Name** : saisissez le nom de la colonne du schéma à insérer, ou le nom de la colonne du schéma utilisée pour remplacer une colonne existante. 
• **SQL expression** : instruction SQL à exécuter pour insérer ou remplacer la colonne correspondante. |
<table>
<thead>
<tr>
<th>Position</th>
<th>Sélectionnez Before, After ou Replace, selon l’action à effectuer sur la colonne de référence.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference column</td>
<td>nom de la colonne de référence pouvant être utilisée pour trouver la nouvelle colonne à insérer ou qui sera remplacée.</td>
</tr>
</tbody>
</table>

**Use field options**

Cochez cette case et, dans la table **Fields options** qui s’affiche, cochez la case de la colonne correspondante pour personnaliser une requête, particulièrement si plusieurs actions sont effectuées sur les données.

- **Key in update** : cochez la case de la colonne correspondante en vous basant sur les données mises à jour.
- **Key in delete** : cochez la case de la colonne correspondante en vous basant sur les données supprimées.
- **Updatable** : cochez cette case si les données de la colonne correspondante peuvent être mises à jour.
- **Insertable** : cochez cette case si les données dans la colonne correspondante peuvent être insérées.

**Enable debug mode**

Cochez cette case pour afficher chaque étape lors du traitement des données dans une base de données.

**Use Batch**

Cochez cette case pour activer le mode de traitement par lots pour le traitement des données et, dans le champ **Batch Size** qui s’affiche, spécifiez le nombre d’enregistrement à traiter dans chaque lot.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Enable parallel execution**

Cochez cette case pour traiter plusieurs flux de données simultanément. Cela permet de traiter les données plus rapidement. Notez que cette fonctionnalité dépend de la capacité de la base de données ou de l’application à gérer plusieurs insertions en parallèle ainsi que le nombre de processeurs utilisés. Lorsque cette case est cochée, vous devez spécifier le nombre d’exécutions parallèles souhaitées, dans le champ **Number of parallel executions**.

**Remarque** : Lorsque l’exécution parallèle est activée, il n’est pas possible d’utiliser des variables globales pour récupérer des valeurs retournées.

**Variables globales**

<p>| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |</p>
<table>
<thead>
<tr>
<th><strong>NB_LINE</strong></th>
<th>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE_INSERTED</strong></td>
<td>Nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong></td>
<td>Nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong></td>
<td>Nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_REJECTED</strong></td>
<td>Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>QUERY</strong></td>
<td>Instruction de requête en cours de traitement. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

**Règle d’utilisation**

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données JDBC. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Scénarios associés

Pour un scénario associé au **tJDBCOutput**, consultez :

- **Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput** à la page 2667 du composant **tMysqlOutput**.
tJDBCRollback

Evite le commit de transaction involontaire en annulant la transaction d’une base de données connectée.

Propriétés du tJDBCRollback Standard

Ces propriétés sont utilisées pour configurer le tJDBCRollback s’exécutant dans le framework de Jobs Standard.

Le composant tJDBCRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Component</td>
<td>Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Cochez cette case pour fermer la connexion à la base de données une fois que le composant a terminé sa tâche. Décochez cette case pour continuer à utiliser la connexion sélectionnée, une fois que le composant a terminé sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics   | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Variables globales

| ERROR_MESSAGE             | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retournne une chaîne de caractères. |
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en association avec des composants JDBC, notamment avec le tJDBCConnection et le tJDBCCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Scénario associé

Pour plus d’informations relatives au fonctionnement du composant tJDBCRollback, consultez tMysqlConnection à la page 2618.
tJDBCRow

Agit sur la structure même de la base de données ou sur les données (mais sans les manipuler) en utilisant le SQLBuilder pour écrire rapidement et aisément vos requêtes.

tJDBCRow est le composant générique pour tout type de base de données utilisant une API JDBC. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tJDBCRow Standard

Ces propriétés sont utilisées pour configurer le tJDBCRow s'exécutant dans le framework de Jobs Standard.

Le composant tJDBCRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• Built-In : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• Repository : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
<tr>
<td>Connection Component</td>
<td>Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant.</td>
</tr>
</tbody>
</table>
### JDBC URL


### Drivers

Renseignez cette table pour charger les pilotes Jar requis. Pour ce faire, cliquez autant de fois que nécessaire sur le bouton `[+]` sous la table, pour ajouter autant de lignes que vous le souhaitez, chaque ligne pour un pilote Jar. Sélectionnez ensuite la cellule et cliquez sur le bouton `[...]` à droite de la cellule, pour ouvrir l’assistant Module dans lequel vous pouvez sélectionner votre pilote Jar. Par exemple, le pilote Jar `RedshiftJDBC41-1.1.13.1013.jar` pour la base de données Redshift.

### Driver Class

Nom de la classe, entre guillemets doubles, pour le pilote spécifié. Par exemple, pour le pilote `RedshiftJDBC41-1.1.13.1013.jar`, le nom à saisir est `com.amazon.redshift.jdbc41.Driver`.

### Use Id et Password

Données d’authentification de l’utilisateur à la base de données.

Pour saisir le mot de passe, cliquez sur le bouton `[...]` à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur Edit schema pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Saisissez le nom de la table à traiter.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query Type et Query</strong></td>
<td>Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.</td>
</tr>
<tr>
<td>• <strong>Built-In</strong></td>
<td>saisissez la requête manuellement dans le champ <strong>Query</strong> ou cliquez sur le bouton [...] à côté du champ <strong>Query</strong> afin de construire l’instruction graphiquement, à l’aide du SQLBuilder.</td>
</tr>
<tr>
<td>• <strong>Repository</strong></td>
<td>sélectionnez la requête correspondante stockée dans le référentiel an cliquant sur le bouton [...]. Dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez la requête à utiliser. Le champ <strong>Query</strong> est automatiquement renseigné.</td>
</tr>
<tr>
<td><strong>Guess Query</strong></td>
<td>Cliquez sur ce bouton pour générer la requête dans le champ <strong>Query</strong> à partir de la table et du schéma définis.</td>
</tr>
<tr>
<td><strong>Specify a data source alias</strong></td>
<td>Cochez cette case et, dans le champ <strong>Data source alias</strong> qui s’affiche, spécifiez l’alias d’une source de données créée du côté Talend Runtime, pour utiliser le pool de connexions partagées défini dans la configuration de la source de données. Cette option fonctionne uniquement lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour un scénario associé, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647. Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l’alias de la source de données spécifié. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste déroulante <strong>Connection Component</strong>.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. Décocochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque des erreurs sont ignorées, vous pouvez collecter les lignes en erreur à l’aide d’un lien <strong>Row &gt; Reject</strong>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| **Propagate QUERY's recordset** | Cochez cette case afin de propager le résultat de la requête au flux de sortie. Dans la liste **use column** affichée, vous devez sélectionner une colonne dans laquelle insérer le résultat de la requête. Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de... |
**Use PreparedStatement**

Cochez cette case si vous souhaitez interroger la base de données à l’aide d’une instruction préparée. Dans la table **Set PreparedStatement Parameters** qui s’affiche, spécifiez la valeur pour chaque paramètre représenté par un point d’interrogation ?, dans l’instruction SQL définie dans le champ **Query**.

- **Parameter Index**: position du paramètre dans l’instruction SQL.
- **Parameter Type**: type de données du paramètre.
- **Parameter Value**: valeur du paramètre.

Pour un scénario utilisant cette propriété, consultez **Scénario 2 : Utiliser l’instance PreparedStatement pour faire une requête sur des données** à la page 2702.

**Commit every**

Spécifiez le nombre de lignes à traiter avant de commiter un lot de lignes dans la base de données.

Cette option assure la qualité de la transaction (mais pas de rollback) et de meilleures performances lors de l’exécution.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

<table>
<thead>
<tr>
<th>Variable</th>
<th>Définition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survenue. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>QUERY</td>
<td>Instruction de requête en cours de traitement. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur toute base de données utilisant une connexion JDBC et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple...</td>
</tr>
</tbody>
</table>
Scénarios associés

Pour un scénario associé, consultez :

- **Scénario : Combiner deux flux pour une sortie sélective** à la page 2706.
- **Procédure** du composant **tDBSQLRow**.
- **Scénario : Supprimer et re-générer un index de table MySQL** à la page 2700 du composant **tMysqlRow**.
tJDBCSCDELT

Suit les modifications de données dans une table de base de données source, à l’aide d’une méthode Slowly Changing Dimensions (SCD) de Type 1 et/ou de Type 2 et écrit les données actuelles et historiques dans une table de dimension SCD spécifiée.

Propriétés du tJDBCSCDELT Standard

Ces propriétés sont utilisées pour configurer le tJDBCSCDELT s’exécutant dans le framework de Jobs Standard.

Le composant tJDBCSCDELT Standard appartient à deux familles : Business Intelligence et Databases. Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |


| Driver JAR | Renseignez cette table afin de charger les Jar des pilotes requis. Pour ce faire, cliquez sur le bouton [+] sous la table, pour ajouter autant de lignes que nécessaire, chaque ligne pour un Jar de pilote. Sélectionnez la cellule et cliquez sur le bouton [...] à droite de la cellule pour ouvrir l’assistant **Select Module**, dans lequel vous pouvez sélectionner le Jar souhaité. Par exemple, le Jar du pilote **RedshiftJDBC41-1.1.13.1013.jar** pour la base de données Redshift. |

| Driver Class | Saisissez entre guillemets doubles le nom de la classe pour le pilote spécifié. Par exemple, pour le pilote **RedshiftJDBC41-1.1.13.1013.jar**, le nom à saisir est `com.amazon.redshift.jdbc41.Driver`. |
**Username et Password**

Saisissez les données d’authentification de l’utilisateur de la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**DB Schema**

Spécifiez le nom du schéma de la base de données. Pour les bases de données ne supportant pas les schémas, laissez le champ vide.

**Source table**

Spécifiez le nom de la table source d’entrée dont les modifications sur les données seront capturées par le SCD.

**Table**

Spécifiez le nom de la table de dimension dans laquelle les données capturées par le SCD seront écrites.

**Action on table**

Sélectionnez une opération à effectuer sur la table définie, vous pouvez effectuer l’une des opérations suivantes :
- **None** : aucune opération n’est effectuée.
- **Drop and create table** : la table est supprimée et créée à nouveau.
- **Create table** : la table n’existe pas et est créée.
- **Create table if does not exist** : la table est créée si elle n’existe pas.
- **Drop table if exist and create** : la table est supprimée si elle existe déjà et créée à nouveau.
- **Clear table** : le contenu de la table est supprimé. Vous pouvez annuler cette opération.
- **Truncate table** : le contenu de la table est supprimé. Vous ne pouvez pas annuler cette opération.

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.
- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th>Surrogate key</th>
<th>Définissez la colonne dans laquelle la clé de substitution sera stockée. Une clé de substitution peut être générée à partir de la méthode sélectionnée dans la liste Creation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation</td>
<td>Sélectionnez l’une des méthodes suivantes pour générer la clé de substitution.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Auto increment</strong> : un entier s’incrémentant automatiquement sera utilisé.</td>
</tr>
<tr>
<td></td>
<td>- <strong>DB sequence</strong> : une séquence sera utilisée.</td>
</tr>
<tr>
<td>Source keys</td>
<td>Spécifiez une ou plusieurs colonnes utilisées comme clés assurant l’unicité des données entrantes.</td>
</tr>
<tr>
<td>Use SCD type 1 fields</td>
<td>Cochez cette case et, dans la table SCD type 1 fields, spécifiez les colonnes dont les valeurs modifiées seront suivies à l’aide du Type 1 de SCD.</td>
</tr>
<tr>
<td>SCD type 1 fields</td>
<td>Cliquez sur le bouton [+] pour ajouter autant de lignes que nécessaire, chaque ligne pour une colonne. Cliquez sur le côté droit de la cellule et dans la liste déroulante, sélectionnez la colonne dont les valeurs modifiées seront suivies à l’aide du Type 1 de SCD.</td>
</tr>
<tr>
<td>Use SCD type 2 fields</td>
<td>Cochez cette case et, dans la table SCD type 2 fields, spécifiez les colonnes dont les valeurs modifiées seront suivies à l’aide du Type 2 de SCD.</td>
</tr>
<tr>
<td>SCD type 2 fields</td>
<td>Cliquez sur le bouton [+] pour ajouter autant de lignes que nécessaire, chaque ligne pour une colonne. Cliquez sur le côté droit de la cellule et dans la liste déroulante, sélectionnez la colonne dont les valeurs modifiées seront suivies à l’aide du Type 2 de SCD.</td>
</tr>
<tr>
<td>Start date</td>
<td>Spécifiez la colonne contenant la date de départ pour le Type 2 de SCD.</td>
</tr>
<tr>
<td>End date</td>
<td>Spécifiez la colonne contenant la date de fin pour le Type 2 de SCD.</td>
</tr>
<tr>
<td>Log active status</td>
<td>Cochez cette case et, dans la liste déroulante Active field qui s’affiche, sélectionnez la colonne contenant la valeur de statut true ou false, ce qui permet d’indiquer l’enregistrement actif pour le Type 2 de SCD.</td>
</tr>
</tbody>
</table>
Suivre des modifications de données dans une table Snowflake à l'aide du composant tJDBCSCDELT

Ce scénario décrit un Job capturant les modifications des données des employés dans une table Snowflake, à l'aide des méthodes SCD (Slowly Changing Dimensions) de Type 1 et de Type 2 implémentée par le composant tJDBCSCDELT et écrivant les données actuelles et historiques dans une table de dimension SCD.

Les données d’entrée contiennent différents détails relatifs aux employés, comme leur nom (name), leur poste (role) ou leur salaire (salary). Une colonne id est ajoutée pour assurer l'unicité des données d’entrée.
Les données suivantes sont insérées dans une nouvelle table Snowflake.

<table>
<thead>
<tr>
<th>#id;name;role;salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>111;Mark Smith;tester;15000.00</td>
</tr>
<tr>
<td>222;Thomas Johnson;developer;18000.00</td>
</tr>
<tr>
<td>333;Teddy Brown;tester;16000.00</td>
</tr>
</tbody>
</table>

La table est ensuite mise à jour avec les données renouvelées des employés.

<table>
<thead>
<tr>
<th>#id;name;role;salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>111;Mark Smith;tester;15000.00</td>
</tr>
<tr>
<td>222;Thomas Johnson;tester;18000.00</td>
</tr>
<tr>
<td>333;Teddy Brown;writer;17000.00</td>
</tr>
<tr>
<td>444;John Clinton;developer;19000.00</td>
</tr>
</tbody>
</table>

Vous pouvez voir que le poste de Thomas Johnson est passé de developer à tester, le poste de Teddy Brown est passé de tester à writer et son salaire est passé de 16000.00 à 17000.00. De plus, un nouvel enregistrement d’ID 444 a été inséré. Dans ce scénario,

- les données relatives au nouveau nom et au nouveau poste doivent écraser les données existantes, vous allez donc effectuer une méthode SCD de Type 1 sur ces données,
- vous souhaitez conserver l’historique complet des données relatives aux salaires, toujours créer un nouvel enregistrement avec les données modifiées et fermer l’enregistrement précédent, vous allez donc effectuer une méthode SCD de Type 2 sur ces données.

Pour plus d’informations concernant les types SCD, consultez Méthodologie de gestion du SCD à la page 2716.

**Créer un Job pour suivre les modifications de données dans une table Snowflake à l’aide du tJDBCSCDELT**

**Procédure**

1. Créez un nouveau Job et ajoutez un composant tJDBCConnection, deux tJDBCRow, deux tFixedFlowInput, deux tJDBCOutput, deux tJDBCSCDELT, deux tJDBCInput et deux tLogRow au Job.
2. Reliez le premier tFixedFlowInput au tJDBCOutput à l'aide d'un lien Row > Main.
3. Répétez l’opération pour relier le premier tJDBCInput au premier tLogRow, le second tFixedFlowInput au second tJDBCOutput et le second tJDBCInput au second tLogRow.

4. Reliez le tDBCCConnection au premier tJDBCRow à l’aide d’un lien Trigger > On Subjob Ok.

5. Répétez l’opération pour relier le premier tJDBCRow au second tJDBCRow, le second tJDBCRow au premier tFixedFlowInput, le premier tFixedFlowInput au premier tJDBCSCDELT, le premier tJDBCSCDELT au premier tDBCInput, le premier tDBCInput au second tFixedFlowInput, le second tFixedFlowInput au second tJDBCSCDELT, le second tJDBCSCDELT au second tDBCInput.

Ouvrir une connexion à une base de données Snowflake

Procédure

1. Double-cliquez sur le tJDBCConnection pour ouvrir sa vue Basic settings.

2. Dans le champ JDBC URL, saisissez la chaîne de caractères de connexion à Snowflake via le pilote JDBC.

   Pour plus d’informations concernant l’utilisation du pilote JDBC pour vous connecter à Snowflake et comment spécifier la chaîne de caractères de connexion, consultez JDBC Driver Connection String (en anglais).

3. Cliquez sur le bouton [+] sous la table Driver JAR pour ajouter une ligne et, dans la cellule Jar name de la nouvelle ligne, saisissez le nom du fichier Jar pour le pilote Snowflake JDBC, snowflake-jdbc-3.2.2.jar, dans cet exemple.


5. Dans les champs Username et Password, saisissez les informations d’authentification.

Créer une table et une séquence Snowflake

Procédure

1. Double-cliquez sur le premier composant tJDBCRow pour ouvrir sa vue Basic settings.

2. Cochez la case Use an existing connection et, dans la liste déroulante Component List qui s’affiche, sélectionnez le composant de connexion duquel réutiliser la connexion créée, tJDBCConnection_1 dans cet exemple.

3. Dans le champ Query, saisissez la commande SQL utilisée pour créer une nouvelle table.
Dans cet exemple, la commande SQL est:

```sql
CREATE OR REPLACE TABLE employee (id INTEGER, name VARCHAR(50), role VARCHAR(50), salary DOUBLE, PRIMARY KEY(id)),
```

créant une nouvelle table `employee` contenant quatre colonnes, `id` de type INTEGER étant la clé primaire, `name` et `role` de type VARCHAR et `salary` de type DOUBLE. Cette table sera utilisée pour stocker les données des employés.

4. Double-cliquez sur le second `tJDBCRow` pour ouvrir sa vue `Basic settings`.

5. Cochez la case `Use an existing connection` et, dans la liste `Component List` qui s’affiche, sélectionnez le composant de connexion duquel réutiliser la connexion créée, `tJDBCConnection_1` dans cet exemple.

6. Dans le champ `Query`, saisissez la commande SQL utilisée pour créer une séquence Snowflake. Dans cet exemple, la commande SQL est:

```sql
create or replace sequence employee_sequence,
```

créant une nouvelle séquence `employee_sequence`. Cette séquence sera utilisée par le composant `tJDBCSCDELT` afin de générer la clé de substitution pour la méthode SCD de Type 2.

**Insérer des données dans la nouvelle table Snowflake**

**Procédure**

1. Double-cliquez sur le premier composant `tFixedFlowInput` pour ouvrir sa vue `Basic settings`.

2. Cliquez sur le bouton `[...]` à côté du champ `Edit schema` et, dans la fenêtre qui s’ouvre, définissez le schéma en ajoutant quatre colonnes: `id` de type Integer étant la clé primaire, `name` et `role` de type String et `salary` de type Double.

3. Cliquez sur `OK` pour sauvegarder les modifications apportées au schéma. Dans la boîte de dialogue qui s’ouvre, cliquez sur `Yes` afin de propager le schéma au composant suivant.

4. Sélectionnez `Use Inline Content` dans la zone `Mode`. Dans le champ `Content` qui s’affiche, saisissez les données suivantes relatives aux employés, données qui seront insérées.

```sql
111;Mark Smith;tester;15000.00
222;Thomas Johnson;developer;18000.00
333;Teddy Brown;tester;16000.00
```

5. Double-cliquez sur le premier `tJDBCOoutput` pour ouvrir sa vue `Basic settings`.
6. Cochez la case **Use an existing connection** et, dans la liste **Component List** qui s’affiche, sélectionnez le composant de connexion afin de réutiliser la connexion créée, `tJDBCConnection_1` dans cet exemple.

7. Dans le champ **Table**, saisissez le nom de la table dans laquelle les données des employés seront écrites, `employee` dans cet exemple.

8. Dans la liste **Action on data**, sélectionnez **Insert** pour insérer les données des employés transférées depuis le premier composant `tFixedFlowInput`.

**Suivre les modifications d’une insertion de données et écrire ces changements dans une table de dimension SCD**

**Procédure**

1. Double-cliquez sur le premier `tJDBCSCDELT` pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Use an existing connection** et, dans la liste **Component List** qui s’affiche, sélectionnez le composant de connexion afin de réutiliser la connexion créée, `tJDBCConnection_1` dans cet exemple.
3. Dans le champ **Source table**, saisissez le nom de la table dont capturer les modifications, *employee* dans cet exemple.

4. Dans le champ **Table**, saisissez le nom de la table de dimension SCD qui stockera les données des employés actuelles et historiques, *employee_scd* dans cet exemple.

5. Sélectionnez **Drop table if exists and create** dans la liste **Action on table** afin de créer la table de dimension SCD.

6. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans la boîte de dialogue, définissez le schéma en ajoutant neuf colonnes : *sk* et *id* de type Integer étant les clés primaires, *name* et *role* de type String, *salary* de type Double, *start_date* et *end_date* de type Date avec le modèle de date (Date Pattern) *yyyy-MM-dd*, ainsi que *active_status* et *version* de type Integer. Cela fait, cliquez sur **OK** afin de sauvegarder les modifications et de fermer la boîte de dialogue.

7. Dans la liste déroulante **Surrogate key**, sélectionnez le nom de la colonne qui sera utilisée comme clé primaire de la table de dimension SCD, *sk* dans cet exemple.

8. Sélectionnez **DB sequence** dans la liste **Creation** et, dans le champ **Sequence** qui s'affiche, saisissez le nom de la séquence Snowflake utilisée pour générer la clé de substitution pour la méthode SCD de Type 2.

9. Cliquez sur le bouton [+ ] sous la table **Source keys** pour ajouter une ligne, cliquez dans la cellule **Name** et sélectionnez la colonne clé de la table source dans la liste déroulante, *id* dans cet exemple.

10. Cochez la case **Use SCD type 1 fields**, cliquez deux fois sur le bouton [+ ] sous la table **SCD type 1 fields** pour ajouter deux lignes. Cliquez dans chaque cellule et, dans la liste déroulante, sélectionnez la colonne sur laquelle effectuer la méthode SCD de Type 1. Dans cet exemple, ces colonnes sont *name* et *role*.

11. Cochez la case **Use SCD type 2 fields**, cliquez sur le bouton [+ ] sous la table **SCD type 2 fields** pour ajouter une ligne. Cliquez dans la cellule et sélectionnez la colonne sur laquelle effectuer la méthode SCD de Type 2. Dans cet exemple, la colonne est *salary*. 

---

2006
12. Dans les listes déroulantes **Start date** et **End date**, sélectionnez les colonnes utilisées pour contenir la date de début et la date de fin pour la méthode SCD de Type 2, respectivement **start_date** et **end_date** dans cet exemple.

13. Cochez la case **Log active status** et, dans la liste **Active field** qui s’affiche, sélectionnez la colonne utilisée pour contenir la valeur du statut actif pour la méthode SCD de Type 2, permettant d’identifier les enregistrements actifs, **active_status** dans cet exemple.

14. Cochez la case **Log versions** et, dans la liste déroulante **Version field**, sélectionnez la colonne utilisée pour contenir le numéro de version des enregistrements pour la méthode SCD de Type 2, **version** dans cet exemple.

15. Sélectionnez **Mapping Snowflake** dans la liste **Mapping** pour utiliser le fichier de mapping des métadonnées Snowflake.

**Récupérer les mises à jour de l’insertion de données depuis la table de dimension SCD**

**Procédure**

1. Double-cliquez sur le premier **tJDBCInput** pour ouvrir sa vue **Basic settings**.
2. Cochez la case **Use an existing connection** et, dans la liste **Component List**, sélectionnez le composant de connexion duquel réutiliser la connexion créée, **tJDBCConnection_1** dans cet exemple.
3. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans la boîte de dialogue, définissez le schéma en ajoutant neuf colonnes : **sk** et **id** de type Integer étant les clés primaires, **name** et **role** de type String, **salary** de type Double, **start_date** et **end_date** de type Date avec le modèle de date (Date Pattern) yyyy-MM-dd, ainsi que **active_status** et **version** de type Integer. Cela fait, cliquez sur **OK** afin de sauvegarder les modifications et de fermer la boîte de dialogue.

Le schéma du premier composant **tJDBCInput** est le même que le schéma du **tJDBCSCDELT**, vous pouvez donc juste le copier-coller.

4. Dans le champ **Query**, saisissez la commande SQL utilisée pour récupérer les données de la table de dimension SCD, **select * from employee_scd** dans cet exemple.
5. Double-cliquez sur le premier **tLogRow** et, dans la zone **Mode** de sa vue **Basic settings**, sélectionnez **Table** pour afficher les données récupérées sous forme de tableau.

**Mettre à jour les données dans la table Snowflake**

**Procédure**

1. Double-cliquez sur le second **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.
2. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans la fenêtre qui s’ouvre, définissez le schéma en ajoutant quatre colonnes : **id** de type Integer étant la clé primaire, **name** et **role** de type String et **salary** de type Double.

Ce schéma est le même que celui du premier **tFixedFlowInput**, vous pouvez donc le copier-coller.

3. Cliquez sur **OK** afin de sauvegarder les modifications du schéma. Dans la boîte de dialogue qui s’ouvre, cliquez sur **Yes** afin de propager le schéma au composant suivant.
4. Sélectionnez Use Inline Content dans la zone Mode. Dans le champ Content qui s’affiche, saisissez les données suivantes des employés, afin de mettre à jour les données existantes.

<table>
<thead>
<tr>
<th>ID</th>
<th>Nom</th>
<th>Rôle</th>
<th>Salaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>Mark Smith</td>
<td>tester</td>
<td>15000.00</td>
</tr>
<tr>
<td>222</td>
<td>Thomas Johnson</td>
<td>tester</td>
<td>18000.00</td>
</tr>
<tr>
<td>333</td>
<td>Teddy Brown</td>
<td>writer</td>
<td>17000.00</td>
</tr>
<tr>
<td>444</td>
<td>John Clinton</td>
<td>developer</td>
<td>19000.00</td>
</tr>
</tbody>
</table>

5. Double-cliquez sur le second tJDBCOutput pour afficher sa vue Basic settings.

6. Cochez la case Use an existing connection et, dans la liste Component List qui s’affiche, sélectionnez le composant de connexion duquel réutiliser la connexion créée, tJDBCConnection_1 dans cet exemple.

7. Dans le champ Table, saisissez le nom de la table dans laquelle mettre à jour les données, employe dans cet exemple.

8. Sélectionnez Insert or update dans la liste Action on data.

**Suivre les modifications des mises à jour et écrire des changements dans une table de dimension SCD**

**Procédure**

1. Double-cliquez sur le second tJDBCSCDELT pour ouvrir sa vue Basic settings.

2. Répétez les étapes 2 à la page 2005 à 15 à la page 2007 de la procédure Suivre les modifications d’une insertion de données et écrire ces changements dans une table de dimension SCD à la page 2005 afin de configurer le second tJDBCSCDELT.

**Récupérer les modifications des mises à jour depuis la table de dimension**

**Procédure**

1. Double-cliquez sur le second tJDBCInput pour ouvrir sa vue Basic settings.

2. Répétez les étapes 2 à la page 2007 à 4 à la page 2007 de la procédure Récupérer les mises à jour de l’insertion de données depuis la table de dimension SCD à la page 2007 afin de configurer le second tJDBCInput.

3. Double-cliquez sur le second tLogRow et, dans la zone Mode de sa vue Basic settings, sélectionnez Table pour afficher les données récupérées sous forme de tableau.

**Exécuter le Job pour suivre les modifications de données dans une table Snowflake à l’aide du tJDBCSCDELT**

**Procédure**

1. Appuyez sur les touches Ctrl + S afin de sauvegarder le Job.

2. Appuyez sur F6 pour exécuter le Job.
Comme affiché ci-dessus, l’ancien poste developer de Thomas Johnson est écrasé par son nouveau poste tester, car une méthode SCD de Type 1 est effectuée sur la colonne role.
Un nouvel enregistrement avec une valeur de clé de substitution définie à 4 est créé pour la modification du salaire de Teddy Brown de 16000.00 à 17000.00, car une méthode SCD de Type 2 a été effectuée sur la colonne salary.
tJDBCSP

Centralise des requêtes multiples ou complexes dans une base de données afin de les appeler plus facilement.

Propriétés du tJDBCSP Standard

Ces propriétés sont utilisées pour configurer le tJDBCSP s’exécutant dans le framework de Jobs Standard.

Le composant tJDBCSP Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td>· Built-In</td>
<td>les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>· Repository</td>
<td>les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
<tr>
<td>Connection Component</td>
<td>Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant.</td>
</tr>
<tr>
<td>Drivers</td>
<td>Renseignez cette table pour charger les pilotes Jar requis. Pour ce faire, cliquez autant de fois que nécessaire.</td>
</tr>
</tbody>
</table>
nécessaire sur le bouton [+] sous la table, pour ajouter autant de lignes que vous le souhaitez, chaque ligne pour un pilote Jar. Sélectionnez ensuite la cellule et cliquez sur le bouton […] à droite de la cellule, pour ouvrir l’assistant Module dans lequel vous pouvez sélectionner votre pilote Jar. Par exemple, le pilote Jar RedshiftJDBC41-1.1.13.1013.jar pour la base de données Redshift.

| Driver Class | Nom de la classe, entre guillemets doubles, pour le pilote spécifié. Par exemple, pour le pilote RedshiftJDBC41-1.1.13.1013.jar, le nom à saisir est com.amazon.redshift.jdbc41.Driver. |
| Use Id et Password | Données d’authentification de l’utilisateur à la base de données. Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| SP Name | Nom de la procédure stockée. |
| Is function | Cochez cette case si la procédure stockée est une fonction et si une valeur est retournée depuis la procédure stockée. Une fois cette case cochée, vous devez sélectionner, dans la liste déroulante Return result in la colonne dans laquelle la valeur retournée sera sauvegardée. |
| Set SP Parameters | Spécifiez les paramètres de la procédure stockée en cliquant sur le bouton [+] pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre et configurez les valeurs suivantes pour chaque paramètre. |
| • Schema Column : colonne contenant la valeur du paramètre. | • Parameter Type : type du paramètre. |
| • IN : paramètre d’entrée. | • OUT : paramètre de sortie/valeur retournée. |
| • IN OUT : combinaison des paramètres d’entrée et de sortie. Cela signifie que le paramètre peut être passé à la procédure stockée, modifié par la procédure et retourné avec une nouvelle valeur. | • RECORDSET : paramètre d’entrée qui sera retourné en tant qu’ensemble de valeurs, plutôt qu’en une seule valeur. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. |
- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.


Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automātiquement en type built-in.

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Specify a data source alias

Cochez cette case et, dans le champ **Data source alias** qui s’affiche, spécifiez l’alias d’une source de données créée du côté Talend Runtime, pour utiliser le pool de connexions partagées défini dans la configuration de la source de données. Cette option fonctionne uniquement lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour un scénario associé, consultez **Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL** à la page 2647.

Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l’alias de la source de données spécifié.

Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste déroulante **Connection Component**.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |
Variables globales

| ERROR_MESSAGE | Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

Utilisation

| Règle d’utilisation | Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d’entrée sont autorisés. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend. |

Limitation | La syntaxe de la Procédure Stockée doit correspondre à celle de la base de données. |

Scénarios associés

Pour un scénario associé, consultez :

- Scénario : Exécuter une procédure stockée à l’aide du tMDMSP à la page 2320 du composant tMysqlSP.
- Scénario : Vérifier le format de numéros à l’aide d’une procédure stockée à la page 2976 du composant tOracleSP.

Consultez également Scénario : Insérer des données dans des tables mère/fille à la page 2620 si vous voulez analyser un ensemble d’enregistrements d’une table de données ou d’une requête SQL.
tJDBCTableList

Liste les noms des tables JDBC grâce aux commandes SELECT et WHERE.
tJDBCTableList effectue une opération d’itération sur toutes les tables d’une base de données, grâce à une connexion JDBC définie.

**Propriétés du tJDBCTableList Standard**

Ces propriétés sont utilisées pour configurer le tJDBCTableList s’exécutant dans le framework de Jobs Standard.
Le composant tJDBCTableList Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Database type</th>
<th>Sélectionnez dans la liste la base de données que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant de connexion tJDBCConnection ou le composant de connexion de la base de données sélectionnée dans la liste Database type.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Use filter</th>
<th>Cochez cette case afin de filtrer les tables sur lesquelles effectuer une itération.</th>
</tr>
</thead>
</table>
| Regular expression for tables name | Saisissez l’expression régulière permettant d’identifier le nom des tables.  
Ce champ est disponible lorsque la case Use filter est cochée. |
| Filter criteria     | Sélectionnez le critère permettant de filtrer le nom des tables.  
**Include** : effectue une itération sur les noms de tables identifiés par l’expression régulière uniquement.  
**Exclude** : effectue une itération sur les noms de tables non identifiés par l’expression régulière.  
Cette liste est disponible lorsque la case Use filter est cochée. |
| tStatCatcher statistics | Cochez cette case afin de collecter les données de log au niveau des composants. |
Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>CURRENT_TABLE : nom de la table sur laquelle se fait l’itération. Cette variable est une variable <strong>Flow</strong> et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>NB_TABLE</strong> : nombre de tables itérées jusqu’à présent. Cette variable est une variable <strong>Flow</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

Utilisation

| Règle d’utilisation | Ce composant est utilisé en association avec les autres composants JDBC, notamment avec le **tJDBCConnection**. |

Scénario associé

Pour un scénario associé, consultez **Scénario : Itérer une table de base de données et lister le nom des colonnes de la table** à la page 2611.
**tJIRAImput**

Récupère depuis JIRA des informations relatives à des tickets grâce à une requête JQL ou des informations relatives à un projet, en se basant sur un ID de projet spécifié.

**Propriétés du tJIRAImput Standard**

Ces propriétés sont utilisées pour configurer le tJIRAImput s’exécutant dans le framework de Jobs Standard.
Le composant tJIRAImput Standard appartient à la famille Business.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Host URL</th>
<th>Spécifiez l’URL utilisée pour accéder à JIRA.</th>
</tr>
</thead>
</table>

| User Id et Password | Spécifiez les données d’authecntification utilisées pour accéder à JIRA.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
|---------------------|---------------------------------------------------|

<table>
<thead>
<tr>
<th>JIRA resource</th>
<th>Sélectionnez la ressource JIRA dans la liste déroulante, ISSUE ou PROJECT.</th>
</tr>
</thead>
</table>

| Schema et Edit schema | Un schéma est une description de lignes. Il définit le nombre de champs (colonnes) à traiter et à passer au composant suivant.  
Notez que le schéma de ce composant est en lecture seule et contient une colonne *json* de type **String**. Il stocke les informations relatives aux tickets ou aux projets au format JSON. Vous pouvez cliquer sur **Edit schema** pour voir le schéma. |
|-----------------------|-----------------------------------------------------------------|

| JQL | Spécifiez la requête JQL (Jira Query Language) sur laquelle baser votre recherche des tickets dans JIRA.  
Ce champ est disponible uniquement lorsque l’option ISSUE est sélectionnée dans la liste déroulante Jira resource. |
|----------------|-------------------------------------------------------------|

| Project ID | Saisissez l’ID du projet dont vous souhaitez récupérer les informations depuis JIRA.  
Ce champ est disponible uniquement lorsque l’option PROJECT est sélectionnée dans la liste déroulante Jira resource. |
|-------------|----------------------------------------------------------|
**Advanced settings**

<table>
<thead>
<tr>
<th>Batch size</th>
<th>Spécifiez le nombre de tickets à traiter dans chaque lot. Ce champ est disponible uniquement lorsque l’option ISSUE est sélectionnée dans la liste déroulante Jira resource.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
<td><strong>NB_LINE</strong></td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Ce composant ne supporte pas la récupération de commentaires des tickets dans JIRA.</td>
</tr>
</tbody>
</table>

**Récupérer les informations de projet depuis l’application JIRA**

Voici un exemple d’utilisation des composants Talend pour récupérer les informations d’un projet depuis l’application JIRA, extraire les informations requises en se basant sur la requête JSONPath, puis écrire les informations extraites dans un fichier JSON.

**Créer un Job pour récupérer les informations du projet depuis l’application JIRA**
Avant de commencer

Un projet avec la clé `DOC` a été créé dans l’application JIRA.

Procédure

1. Créez un nouveau Job et ajoutez un composant `tJIRAInput`, un `tExtractJSONFields` et un `tFileOutputJSON` dans le Job.
2. Reliez le `tJIRAInput` au `tExtractJSONFields` à l’aide d’un lien `Row > Main`.
3. Reliez le composant `tExtractJSONFields` au `tFileOutputJSON` à l’aide d’un lien `Row > Main`.

Configurer le Job pour récupérer les informations du projet depuis l’application JIRA

Procédure

1. Double-cliquez sur le `tJIRAInput` pour ouvrir sa vue `Basic settings`.

   ![tJIRAInput_1](image)

   2. Dans les champs `Host URL`, `User Id` et `Password`, spécifiez les informations d’authentification requises pour accéder à l’application JIRA.
   4. Double-cliquez sur le composant `tExtractJSONFields` pour ouvrir sa vue `Basic settings`. 
5. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans la fenêtre, définissez le schéma en ajoutant six colonnes de type String, **id**, **key**, **name**, **description**, **assigneeType** et **ProjectTypeKey**, qui contiendront les informations correspondantes du projet spécifié.

Cela fait, cliquez sur **OK** afin de sauvegarder vos modifications et fermer la boîte de dialogue.

6. Sélectionnez le champ JSON à extraire, dans la liste déroulante **JSON field**. Dans cet exemple, saisissez **json**, champ passé du composant **tJIRAInput**.

7. Dans le champ **Loop Jsonpath query**, spécifiez le chemin vers le nœud sur lequel la boucle se base. Dans cet exemple, saisissez **$**, qui est le nœud racine du champ JSON.

8. Dans la table **Mapping**, les cellules de la colonne **Column** sont automatiquement renseignées avec les colonnes du schéma. Vous devez spécifier le nœud JSON correspondant contenant les données souhaitées pour chaque colonne du schéma. Dans cet exemple, spécifiez **$.id** pour la colonne **id**, **$.key** pour la colonne **key**, **$.name** pour la colonne **name** pour la colonne, **$.description** pour la colonne **description**, **$.assigneeType** pour la colonne assigneeType et **$.projectTypeKey** pour la colonne **projectTypeKey**.

9. Double-cliquez sur le **tFileOutputJSON** pour ouvrir sa vue **Basic settings**.

10. Dans le champ **File Name**, spécifiez le chemin d'accès au fichier dans lequel les informations extraites du projet seront écrites, D:/JiraComponents/project_info.json dans cet exemple. Dans le champ **Name of data block**, saisissez le nom du bloc de données, **project** dans cet exemple.

**Exécuter le Job pour récupérer des informations de projet depuis une application JIRA**

**Procédure**

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.
3. Ouvrez le fichier JSON file **project_info.json** généré.
Notez que toutes les informations extraites du projet sont affichées sur une ligne, dans le fichier JSON généré. Vous pouvez utiliser un éditeur pour formater et indenter les données d’une manière plus organisée, pour un affichage optimal du résultat.

Comme affiché ci-dessus, les informations souhaitées du projet ont bien été extraites et écrites dans le fichier JSON.
tJIRAOoutput

Insère, met à jour ou supprime des informations de projets ou de tickets dans JIRA.

**Propriétés du tJIRAOoutput Standard**

Ces propriétés sont utilisées pour configurer le tJIRAOoutput s'exécutant dans le framework de Jobs Standard.

Le composant tJIRAOoutput Standard appartient à la famille Business.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Host URL</th>
<th>Spécifiez l'URL utilisée pour accéder à JIRA.</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Id et Password</td>
<td>Spécifiez les données d'authentification utilisées pour accéder à JIRA. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>JIRA resource</td>
<td>Sélectionnez la ressource JIRA dans la liste déroulante, ISSUE ou PROJECT.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes. Il définit le nombre de champs (colonnes) à traiter et à passer au composant suivant. Notez que le schéma de ce composant est en lecture seule et contient une colonne json de type String. Il stocke les informations relatives aux tickets ou aux projets au format JSON. Vous pouvez cliquer sur Edit schema pour voir le schéma.</td>
</tr>
<tr>
<td>Output Action</td>
<td>Sélectionnez dans la liste une opération à effectuer. • INSERT : insérer de nouvelles informations relatives à des tickets ou des projets dans JIRA. • UPDATE : modifier des informations relatives à des tickets ou projets dans JIRA. • DELETE : supprimer des données relatives à des tickets ou projets dans JIRA correspondant au flux d'entrée.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| Delete subtasks | Cochez cette case pour supprimer la (les) sous-tâche(s). Cette case est disponible uniquement lorsque l'option ISSUE est sélectionnée dans la liste déroulante JIRA. |
**tStatCatcher Statistics**

Cochez cette case pour collecter les métriques de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.

---

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_SUCCESS : nombre de lignes traitées avec succès. Cette variable est une variable After et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td>NB_REJECT : nombre de lignes rejetées. Cette variable est une variable After et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l'exécution d’un composant. Une variable After fonctionne après l'exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l'aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

---

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de fin dans un Job ou un sous-job et nécessite un lien d’entrée. |

---

**Créer un ticket dans l’application JIRA**

Voici un exemple d’utilisation des composants Talend pour créer un nouveau ticket dans l’application JIRA, récupérer les informations de ce ticket depuis l’application JIRA, puis extraire les informations nécessaires en se basant sur une requête JSONPath et écrire les informations extraites dans un fichier JSON.

**Configurer un Job pour créer un ticket dans l’application JIRA**
Avant de commencer

Un projet avec une clé **DOC** doit avoir été créé dans l’application JIRA et un ticket avec la clé **DOC-1** a été créé dans ce projet.

**Procédure**

1. Créez un nouveau Job et ajoutez un composant **tFileInputDelimited**, un **tJIRAOutput**, un **tJIRAInput**, un **tExtractJSONFields** et un **tFileOutputJSON** dans le Job.
2. Double-cliquez sur le **tJIRAOutput** pour ouvrir sa vue **Basic settings**.
3. Sélectionnez **Issue** dans la liste déroulante **JIRA resource** et **Insert** dans la liste déroulante **Output Action** pour insérer un ticket.
4. Reliez le **tFileInputDelimited** au **tJIRAOutput** à l’aide d’un lien **Row > Main**. Dans la boîte de dialogue qui s’ouvre, cliquez sur **Yes** pour que le **tFileInputDelimited** récupère le schéma du composant **tJIRAOutput**.
5. Répétez l’opération pour relier le **tJIRAInput** au **tExtractJSONFields** et le **tExtractJSONFields** au **tFileOutputJSON**.
6. Reliez le **tFileInputDelimited** au **tJIRAInput** à l’aide d’un lien **Trigger > On Subjob Ok**.

**Créer un nouveau ticket dans l’application JIRA**

**Procédure**

1. Double-cliquez sur le **tFileInputDelimited** pour ouvrir sa vue **Basic settings**.
2. Dans le champ **File name/Stream**, spécifiez le chemin vers le fichier JSON utilisé pour créer le ticket.

Dans cet exemple, un fichier JSON simple `D:/JiraComponents/issue_create.json` sera utilisé pour insérer un nouveau ticket dans le projet dont la clé est **DOC**. Les données du fichier se présentent comme suit :

```json
{"fields": {
  "issuetype": {
    "id": "10001",
    "name": "Task"
  },
  "project": {
    "key": "DOC",
    "name": "Documentation"
  }
},
```
Notez que le composant tJIRAOutput supporte uniquement, pour le moment, la création de projets et tickets dans l’application JIRA, avec un fichier JSON contenant une seule ligne. Vous devez donc joindre toutes les lignes du fichier en une, avant d’exécuter le Job.

Pour plus d’informations concernant les types de tickets, consultez la documentation officielle de JIRA : Issue types (en anglais).

Pour plus d’informations concernant comment trouver l’ID des types de tickets, consultez la documentation officielle de JIRA : Finding the Id for Issue Types (en anglais).

3. Double-cliquez sur le composant tJIRAOutput pour ouvrir sa vue Basic settings.

4. Dans les champs Host URL, User Id et Password, spécifiez les informations d’authentification utilisateur requises pour accéder à l’application JIRA.

Récupérer les informations du nouveau ticket depuis l’application JIRA

Procédure

1. Double-cliquez sur le composant tJIRAInput pour ouvrir sa vue Basic settings.

2. Sélectionnez Issue dans la liste déroulante JIRA resource.

3. Dans le champ JQL, saisissez la requête JQL utilisée pour récupérer les tickets, dans l’application JIRA.

Dans cet exemple, la requête est project = DOC ORDER BY created DESC. Tous les tickets du projet DOC seront récupérés et triés par date de création, en ordre descendant.

4. Double-cliquez sur le composant tExtracJSONFields pour ouvrir sa vue Basic settings.
5. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la fenêtre de schéma qui s'affiche, définissez le schéma en ajoutant cinq colonnes de type String : id, key, project, description et summary, qui contiendront les informations correspondantes pour chaque ticket récupéré. Cela fait, cliquez sur OK pour sauvegarder les modifications et fermer la fenêtre.

6. Sélectionnez le champ JSON à extraire, dans la liste déroulante JSON field. Dans cet exemple, sélectionnez json, passé depuis le composant tJIRAInput.

7. Dans le champ Loop Jsonpath query, spécifiez le chemin vers le nœud sur lequel se base la boucle. Dans cet exemple, spécifiez $, nœud racine du champ JSON.

8. Dans la table Mapping, les cellules de la colonne Column sont automatiquement renseignées par les colonnes du schéma. Vous devez spécifier le nœud JSON correspondant contenant les données souhaitées pour chaque colonne du schéma. Dans cet exemple, $.id pour la colonne id, $.key pour la colonne key, $.fields.project.key pour la colonne project, $.fields.description pour la colonne description et $.fields.summary pour la colonne summary.


10. Dans le champ File Name, spécifiez le chemin vers le fichier dans lequel sont écrites les informations des tickets, D:/JiraComponents/issue_info.json dans cet exemple. Dans le champ Name of data block, saisissez le nom du bloc de données, issue dans cet exemple.

Exécuter le Job pour créer un ticket dans l’application JIRA

Procédure

1. Appuyez sur les touches Ctrl + S pour sauvegarder votre Job.
2. Appuyez sur F6 pour exécuter le Job.
3. Ouvrez le fichier JSON issue_info.json généré.

Notez que toutes les informations extraites des tickets sont affichées dans le fichier JSON généré. Vous pouvez utiliser un outil d’édition pour formater et indenter les données d’une manière plus organisée, pour un affichage optimal des résultats.
Comme affiché ci-dessus, un nouveau ticket avec une clé DOC-2 a été créé et les informations souhaitées pour chaque ticket du projet DOC sont bien extraites et écrites dans le fichier JSON.

Mettre à jour un ticket dans l'application JIRA

Voici un exemple d’utilisation des composants Talend pour mettre à jour un ticket existant dans l’application JIRA, récupérer les informations du ticket mis à jour depuis cette application, puis extraire les informations nécessaires en se basant sur la requête JSONPath et écrire les informations extraites dans un fichier JSON.

Créer un Job pour mettre à jour un ticket dans l’application JIRA

Avant de commencer

Un projet avec la clé DOC a été créé dans l’application JIRA et un ticket avec la clé DOC-2, comme affiché ci-dessous est créé dans ce projet.
Procédure

1. Créez un nouveau Job et ajoutez un composant tFileInputDelimited, un tJIRAOutput, un tJIRAInput, un tExtractJSONFields et un tFileOutputJSON au Job.
2. Double-cliquez sur le composant tJIRAOutput pour ouvrir sa vue Basic settings.
4. Reliez le composant tFileInputDelimited au tJIRAOutput à l’aide d’un lien Row > Main. Dans la boîte de dialogue qui s’ouvre, cliquez sur Yes pour que le tFileInputDelimited récupère le schéma du composant tJIRAOutput.
5. Répétez l’opération pour relier le tJIRAInput au tExtractJSONFields et le tExtractJSONFields au tFileOutputJSON.
6. Reliez le composant tFileInputDelimited au tJIRAInput à l’aide d’un lien Trigger > On Subjob Ok.

Mettre à jour un ticket dans l’application JIRA

Procédure

1. Double-cliquez sur le composant tFileInputDelimited pour ouvrir sa vue Basic settings.
2. Dans le champ File name/Stream, spécifiez le chemin d’accès au fichier JSON utilisé pour mettre à jour le ticket.

Dans cet exemple, un fichier simple JSON D:/JiraComponents/issue_update.json sera utilisé pour mettre à jour un ticket existant avec la clé DOC-2, dans le projet DOC. Les données dans ce fichier se présentent comme suit :

```json
DOC-2;
{"fields": {
    "description": "Update an issue under the DOC project",
    "summary": "Add the documentation for new components tJIRAInput and tJIRAOutput"
}}
```

Notez que le composant tJIRAOutput supporte uniquement, pour le moment, la mise à jour des projets et tickets dans l’application JIRA avec un fichier contenant une ligne unique. Vous devez donc joindre toutes les lignes du fichier pour n’en former plus qu’une, avant d’exécuter le Job.
3. Double-cliquez sur le composant tJIRAOutput pour ouvrir sa vue Basic settings.
4. Dans les champs Host URL, User Id et Password, spécifiez les informations d’authentification requises pour accéder à l’application JIRA.

Récupérer les informations du ticket mis à jour depuis l’application JIRA
**Procédure**

1. Double-cliquez sur le composant **tJIRAInput** pour ouvrir sa vue **Basic settings**.

2. Sélectionnez **Issue** dans la liste déroulante **JIRA resource**.

3. Dans le champ **JQL**, saisissez la requête JQL utilisée pour récupérer le ticket dans l’application JIRA. Dans cet exemple, la requête est `project = DOC AND issuekey = DOC-2`. Le ticket dont la clé est `DOC-2`, dans le projet `DOC` sera récupéré.

4. Double-cliquez sur le composant **tExtractJSONFields** pour ouvrir sa vue **Basic settings**.

5. Cliquez sur le bouton `...` à côté du champ **Edit schema** et dans la fenêtre de schéma qui s’ouvre, définissez le schéma en ajoutant cinq colonnes de type String, `id`, `key`, `project`, `description` et `summary`, qui contiendront les informations correspondantes pour chaque ticket récupéré.

   Cela fait, cliquez sur **OK** pour sauvegarder les modifications et fermer la fenêtre.

6. Sélectionnez le champ JSON à extraire, dans la liste déroulante **JSON field**. Dans cet exemple, sélectionnez `json`, champ passé depuis le composant **tJIRAInput**.

7. Dans le champ **Loop Jsonpath query**, spécifiez le chemin d’accès au nœud sur lequel se base la boucle. Dans cet exemple, spécifiez `$`, nœud racine du champ JSON.

8. Dans la table **Mapping**, les cellules de la colonne **Column** sont automatiquement renseignées avec les colonnes du schéma. Spécifiez le nœud JSON correspondant contenant les données souhaitées pour chaque colonne du schéma. Dans cet exemple, spécifiez `$..id` pour la colonne `id`, `$..key` pour
la colonne key, $.fields.project.key pour la colonne project, $.fields.description pour la colonne description et $.fields.summary pour la colonne summary.


10. Dans le champ File Name, spécifiez le chemin d'accès au fichier dans lequel les informations extraites du ticket seront écrites, D:/JiraComponents/issue_info_update.json dans cet exemple. Dans le champ Name of data block, saisissez le nom du bloc de données, issue dans cet exemple.

Exécuter le Job pour mettre à jour un ticket dans l’application JIRA

Procédure

1. Appuyez sur les touches Ctrl + S pour sauvegarder votre Job.
2. Appuyez sur F6 pour l'exécuter.
3. Ouvrez le fichier JSON issue_info_update.json généré.

Notez que toutes les informations extraites du ticket sont affichées sur une seule ligne dans le fichier JSON généré. Vous pouvez utiliser un éditeur pour formater et indenter les données d'une manière plus organisée, pour un meilleur affichage des résultats.

Comme affiché ci-dessus, le ticket dont la clé est DOC-2 a été mis à jour. Les informations souhaitées pour ce ticket ont bien été extraites et écrites dans le fichier JSON.
**tJMSInput**

Ce composant crée une interface entre une application Java et un système Middleware orienté Message.

Le tJMSInput permet de mettre en place des interactions entre composants applicatifs dans un cadre faiblement couplé, asynchrone et fiable via un serveur JMS.

**Propriétés du tJMSInput Standard**

Ces propriétés sont utilisées pour configurer le tJMSInput s'exécutant dans le framework de Jobs Standard.

Le composant tJMSInput Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Module List</th>
<th>Sélectionnez dans cette liste la bibliothèque à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Factory JDNI Name</td>
<td>Saisissez le nom JDNI.</td>
</tr>
<tr>
<td>Use Specified User Identity</td>
<td>Si vous devez vous identifier, cochez la case et saisissez votre nom d’utilisateur et votre mot de passe. Pour saisir le mot de passe, cliquez sur le bouton <code> [...]</code> à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Enable Durable Subscription</td>
<td>Cochez cette case pour activer la souscription durable.</td>
</tr>
<tr>
<td>ClientID</td>
<td>Saisissez l’ID client pour la souscription durable. Ce champ est disponible uniquement lorsque la case Enable Durable Subscription est cochée.</td>
</tr>
<tr>
<td>Subscriber Name</td>
<td>Saisissez le nom du souscripteur pour la souscription durable. Ce champ est disponible uniquement lorsque la case Enable Durable Subscription est cochée.</td>
</tr>
<tr>
<td>Use JNDI Name Lookup Destination</td>
<td>Cochez cette case pour chercher une destination avec son nom JNDI.</td>
</tr>
<tr>
<td>Message Type</td>
<td>Sélectionnez le type de message, <strong>Topic</strong> ou <strong>Queue</strong>.</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Message From</td>
<td>Saisissez la source du message, telle qu'attendue par le serveur. Celle-ci peut être composé du type et du nom de la source, par exemple : queue/A ou topic/testtopic. Notez que ce champ est sensible à la casse.</td>
</tr>
<tr>
<td>Timeout for Next Message (in sec)</td>
<td>Renseignez le délai (en secondes) avant de passer au message suivant.</td>
</tr>
<tr>
<td>Maximum Messages</td>
<td>Saisissez le nombre maximal de messages à traiter.</td>
</tr>
<tr>
<td>Message Selector Expression</td>
<td>Spécifiez votre filtre.</td>
</tr>
<tr>
<td>Processing Mode</td>
<td>Sélectionnez le mode de traitement des messages : <strong>Raw Message</strong> : message brut, <strong>Message Content</strong> : contenu du message.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs. Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur <strong>Edit schema</strong> afin de visualiser le schéma.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Properties</th>
<th>Cliquez sur le bouton [+ ] sous le tableau afin d'ajouter des lignes contenant le nom et le mot de passe de l'utilisateur requis à l'authentification.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| Global Variables                                   | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette... |
|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette... |
liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début. Il requiert un composant de sortie. |

Scénario associé

Pour un scénario associé, consultez Scénario : Mettre un message dans une file d’attente du serveur ActiveMQ et le retirer de cette file à la page 2035.
**tJMSOutput**

Ce composant crée une interface entre une application Java et un système Middleware orienté Message.

Le tJMSOutput permet de mettre en place des interactions entre composants applicatifs dans un cadre faiblement couplé, asynchrone et fiable via un serveur JMS.

**Propriétés du tJMSOutput Standard**

Ces propriétés sont utilisées pour configurer le tJMSOutput s’exécutant dans le framework de Jobs Standard.

Le composant tJMSOutput Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Module List</th>
<th>Sélectionnez dans cette liste la bibliothèque à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context Provider</td>
<td>Saisissez l’URL de contexte, par exemple com.tibco.tibjms.naming.TibjmsInitialContextFactory. Attention cependant, la syntaxe peut varier selon le serveur JMS utilisé</td>
</tr>
<tr>
<td>Connection Factory JDNI Name</td>
<td>Saisissez le nom JDNI.</td>
</tr>
<tr>
<td>Use Specified User Identity</td>
<td>Si vous devez vous identifier, cochez la case et saisissez votre nom d’utilisateur et votre mot de passe. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Message Type</td>
<td>Sélectionnez le type de message, Topic ou Queue.</td>
</tr>
<tr>
<td>To</td>
<td>Saisissez la cible du message, telle qu’attendue par le serveur.</td>
</tr>
<tr>
<td>Processing Mode</td>
<td>Sélectionnez le mode de traitement des messages : Raw Message : message brut, Message Content : contenu du message.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma du composant tJMSInput est en lecture seule. Il est composé d’une colonne : Message.</td>
</tr>
</tbody>
</table>
**Advanced settings**

| **Delivery Mode** | Sélectionnez un mode de réception dans la liste afin d’assurer la qualité des données lors de la réception :
|                  | **Not Persistent** : Ce mode autorise les pertes de données durant l’échange.
|                  | **Persistent** : Ce mode assure l’intégrité du message à la réception. |

| **Properties** | Cliquez sur le bouton [+ ] sous le tableau afin d’ajouter des lignes contenant le nom et le mot de passe de l’utilisateur requis à l’authentification. |

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| **Global Variables** | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. |
|                     | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**. |

**Utilisation**

| **Règle d’utilisation** | Ce composant est généralement utilisé en tant que composant de sortie. Il requiert un composant d’entrée. |
|                        | Assurez-vous que le serveur JMS correspondant est lancé.
|                        | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet Modules de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation.
Scénario : Mettre un message dans une file d'attente du serveur ActiveMQ et le retirer de cette file

Dans ce scénario, le composant JMSOutput envoie un message dans une file du serveur ActiveMQ, qui est ensuite récupéré par le JMSInput. Ce message est affiché dans la console, via le tLogRow.

Relier les composants

Procédure
1. De la Palette, déposez un tFixedFlowInput, un JMSOutput, un JMSInput et un tLogRow dans l'espace de modélisation graphique.
2. Reliez le tFixedFlowInput au JMSOutput à l’aide d’un lien Row > Main.
4. Connectez le JMSInput au tLogRow à l’aide d’un lien Row > Main.

Configurer les composants

Procédure
1. Double-cliquez sur le tFixedFlowInput pour ouvrir sa vue Basic settings.
Sélectionnez **Use Inline Content (delimited file)** dans la zone **Mode**.

Dans le champ **Content**, saisissez le contenu du message à envoyer au serveur ActiveMQ :

```
message transferred
```

2. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur de schéma.

3. Cliquez sur le bouton [+] pour ajouter une colonne, que vous nommez **messageContent**, de type String.

   Cliquez sur **OK** pour valider la configuration et fermer l’éditeur.

4. Une boîte de dialogue s’ouvre et vous propose de propager le schéma.

   Cliquez sur **Yes** afin de propager le schéma au composant suivant.

5. Double-cliquez sur le **JMSOutput** pour ouvrir sa vue **Basic settings**.

```
Basic settings
```

```
Advanced settings
```

```
Dynamic settings
```

```
View
```

```
Documentation
```

```
Module List
```

```
Context Provider
```

```
Server URL
```

```
Connection Factory JNDI Name
```

```
Use Specified User Identity
```

```
User Name
```

```
Message Type
```

```
To
```

```
Processing Mode
```

```
```
6. Dans la liste Module List, sélectionnez la bibliothèque à utiliser, le Jar `activemq` dans cet exemple.
8. Dans le champ Server URL, saisissez l'URI du serveur ActiveMQ.
9. Dans le champ Connection Factory JNDI Name, saisissez le nom JDNI, "QueueConnectionFactory" dans cet exemple.
10. Cochez la case Use Specified User Identity afin d'afficher les champs User Name et Password, dans lesquels vous pouvez saisir respectivement votre identifiant et votre mot de passe.
11. Dans la liste Message type, sélectionnez Queue.
12. Dans la liste Processing Mode, sélectionnez Message Content.
13. Configurez le JMSInput de la même manière.

Exécuter le Job

Procédure
1. Appuyez sur les touches Ctrl + S pour sauvegarder le Job.
2. Appuyez sur F6 pour exécuter le Job. Le serveur ActiveMQ a démarré à l'adresse suivante :

Dans la console, vous pouvez constater que le message est bien transféré et affiché.

Scénario associé

Pour des scénarios associés, consultez Scénario 1 : Communication asynchrone via un serveur MOM à la page 2416 et Scénario 2 : Transmettre des fichiers XML via un serveur MOM à la page 2419.
tJoin

Ce composant effectue des jointures Inner Join et Outer Join entre le flux de données principal et le flux de référence (Lookup).

Le tJoin joint deux tables, en appliquant la correspondance exacte dans plusieurs colonnes. Il compare les colonnes du flux principal et les colonnes de référence du flux de référence (Lookup), et écrit les données du flux principal et/ou les données rejetées.

Propriétés du tJoin Standard

Ces propriétés sont utilisées pour configurer le tJoin s’exécutant dans le framework de Jobs Standard. Le composant tJoin Standard appartient à la famille Processing. Le composant de ce framework est toujours disponible.

Basic settings

| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
| • View schema : sélectionnez cette option afin de voir le schéma.  
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| **Built-in** : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| **Include lookup columns in output** | Cochez cette case pour inclure les colonnes de référence (lookup), que vous avez définies, dans le flux de sortie. |
### Input key attribute

Sélectionnez la ou les colonne(s) du flux principal qui doivent être vérifiées par rapport à la colonne clé de référence (lookup).

### Lookup key attribute

Sélectionnez les colonnes clé de référence (lookup) que vous utiliserez comme référence, afin de les comparer aux colonnes du flux d’entrée.

### Inner join (with reject output)

Cochez cette case pour joindre d’abord les deux tables et ensuite rassembler les données rejetées du flux principal.

---

### Advanced settings

#### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

---

### Global Variables

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

---

### Utilisation

**Règle d’utilisation**

Ce composant n’est pas un composant de début (fond vert), il nécessite deux composants d’entrée, et un ou plusieurs composant(s) de sortie.

---

**Scénario 1 : Faire une correspondance exacte entre deux colonnes et écrire les données rejetées**

Ce scénario décrit un Job à cinq composants dont le but est d’effectuer une correspondance exacte entre la colonne `firstnameClient` d’un fichier d’entrée, par rapport aux données du fichier de référence d’entrée, et la colonne `lastnameClient`, par rapport aux données du fichier d’entrée de référence. Les sorties de cette correspondance sont écrites dans des fichiers de sortie distincts : les données exactes...
sont écrites dans un fichier Excel, tandis que les données inexactes sont écrites dans un fichier délimité.

Dans ce scénario, vous avez déjà stocké les schémas d'entrée et les fichiers de référence dans le Repository. Pour plus d'informations à propos du stockage de métadonnées dans l'arborescence du Repository, consultez le Guide utilisateur du Studio Talend.

**Déposer et relier les composants**

**Procédure**

1. Dans l’arborescence du Repository, développez le nœud Metadata ainsi que le nœud des fichiers (File) où vous avez stocké les schémas d’entrée, et glissez la métadonnée correspondante dans l’espace de modélisation graphique.
   La boîte de dialogue [Components] s’ouvre.

2. Sélectionnez tFileInputDelimited dans la liste et cliquez sur OK afin de fermer la boîte de dialogue.
   Le composant tFileInputDelimited apparaît dans l’espace de modélisation. Le fichier d’entrée utilisé dans ce scénario est appelé ClientSample. Il contient quatre colonnes, y compris les deux colonnes firstnameClient et lastnameClient sur lesquelles vous voulez appliquer la correspondance exacte.

3. Répétez l’opération pour le deuxième fichier d’entrée que vous souhaitez utiliser comme référence, ClientSample_Update dans ce scénario.

5. Connectez les fichiers principal et de référence au **tJoin** à l’aide d’une connexion de type Row > Main. Le lien principal entre le fichier de référence et le **tJoin** s’affiche comme un lien Lookup (de référence) dans l’espace de modélisation.

6. Connectez le **tJoin** au **tFileOutputExcel** à l’aide d’une connexion Main, et le **tJoin** au **tFileOutputDelimited** avec une connexion Inner join reject.

### Configurer les composants

**Procédure**

1. Au besoin, double-cliquez sur les fichiers d’entrée principal et de référence afin d’afficher leur vue Basic settings. Toutes les propriétés sont automatiquement renseignées. Si vous ne définissez pas vos fichiers d’entrée dans le Repository, remplissez les champs manuellement, après avoir sélectionné Built-in dans le champ Property Type.

   Pour plus d’informations, consultez le Guide utilisateur du Studio Talend.

2. Double-cliquez sur le **tJoin** afin d’afficher sa vue Basic settings et définir ses propriétés.

   ![tJoin Basic settings](image)

   3. Cliquez sur le bouton situé à côté du champ Edit schema pour ouvrir une boîte de dialogue qui affiche la structure des données des fichiers d’entrée, et définissez les données que vous souhaitez passer aux composants de sortie, trois colonnes dans ce scénario, idClient, firstnameClient et lastnameClient.
4. Cliquez sur **OK** pour fermer la boîte de dialogue.

5. Dans la zone **Key definition** de l’onglet **Basic settings** du composant **tJoin**, cliquez sur le bouton [*+*] pour ajouter deux colonnes à la liste, puis sélectionnez les colonnes d’entrée et de sortie sur lesquelles la correspondance exacte sera appliquée, depuis les listes **Input key attribute** et **Lookup key attribute**, respectivement **firstnameClient** et **lastnameClient** dans cet exemple.

6. Cochez la case **Inner join (with reject output)** pour définir l’une des sorties comme table de rejet de l’inner join.

7. Double-cliquez sur le composant **tFileOutputExcel** pour afficher l’onglet **Basic settings** de sa vue **Component**, et définir ses propriétés.

8. Paramétrez le répertoire de destination ainsi que le nom de la feuille (**Sheet name**) et cochez la case **Include header**.
Sauvegarder et exécuter le Job

Procédure

1. Double-cliquez sur le composant **tFileOutputDelimited** pour afficher l'onglet **Basic settings** de sa vue **Component**, et définir ses propriétés.

2. Paramétrez le répertoire de destination ainsi que les séparateurs de lignes (**Row separator**) et de champs (**Field separator**) et cochez la case **Include header**.

3. Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

La sortie de la correspondance exacte des colonnes *firstnameClient* et *lastnameClient* est écrite dans le fichier Excel défini.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>idClient</td>
<td>firstnameClient</td>
</tr>
<tr>
<td>2</td>
<td>28</td>
<td>Herbert</td>
</tr>
<tr>
<td>3</td>
<td>56</td>
<td>Chester</td>
</tr>
</tbody>
</table>

Les données qui ne correspondent pas sont écrites dans le fichier délimité défini.
<table>
<thead>
<tr>
<th>id</th>
<th>Client: firstnameClient: lastnameClient</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dwight: Madison</td>
</tr>
<tr>
<td>2</td>
<td>Franklin: Jackson</td>
</tr>
<tr>
<td>3</td>
<td>Ronald: Buchanan</td>
</tr>
<tr>
<td>4</td>
<td>Bill: Cleveland</td>
</tr>
<tr>
<td>5</td>
<td>William: Harrison</td>
</tr>
<tr>
<td>6</td>
<td>William: Fillmore</td>
</tr>
<tr>
<td>7</td>
<td>Harry: Adams</td>
</tr>
<tr>
<td>8</td>
<td>Harry: McKinley</td>
</tr>
<tr>
<td>9</td>
<td>Herbert: Reagan</td>
</tr>
<tr>
<td>10</td>
<td>Lyndon: Jefferson</td>
</tr>
<tr>
<td>11</td>
<td>Bill: Jackson</td>
</tr>
<tr>
<td>12</td>
<td>John: Hayes</td>
</tr>
<tr>
<td>13</td>
<td>Ulysses: Reagan</td>
</tr>
</tbody>
</table>
**tKafkaCommit**

Ce composant sauvegarde l’état actuel du tKafkaInput auquel il est connecté.

Le composant tKafkaCommit se connecte à un tKafkaInput donné pour effectuer un commit d’un offset d’un consommateur. Notez que le terme “commit” dans ce composant signifie sauvegarder les messages consommés par le tKafkaInput au moment du commit.

**Propriétés du tKafkaCommit Standard**

Ces propriétés sont utilisées pour configurer le tKafkaCommit s’exécutant dans le framework de Jobs Standard.

Le composant tKafkaCommit Standard appartient à la famille Internet.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Commit offsets from | Sélectionnez le composant tKafkaInput duquel les messages consommés sont commités. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case afin de collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

**Utilisation**

| Règle d’utilisation | Même si le tKafkaCommit peut être utilisé comme composant de fin dans le flux d’un sous-job ou être appelé indépendamment comme sous-job, il est généralement utilisé en standalone pour commiter un lot d’offsets en une seule fois. Si vous devez commiter des offsets régulièrement, il est recommandé d’utiliser la fonctionnalité d’Auto-commit dans le tKafkaInput. |

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tKafkaConnection

Ce composant ouvre une connexion Kafka réutilisable.
Le tKafkaConnection ouvre une connexion à un cluster Kafka donné afin que les autres composants Kafka dans les sous-jobs puissent réutiliser cette connexion.

**Propriétés du tKafkaConnection Standard**

Ces propriétés sont utilisées pour configurer le tKafkaConnection s’exécutant dans le framework de Jobs Standard.
Le composant tKafkaConnection Standard appartient à la famille Internet.
Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Version</th>
<th>Sélectionnez la version du cluster Kafka à utiliser.</th>
</tr>
</thead>
</table>

**Zookeeper quorum list**

Saisissez l’adresse du service Zookeeper pour le cluster Kafka à utiliser.
L’adresse doit se présenter sous la forme suivante : *hostname:port*. Ces informations contiennent le nom et le port du nœud hébergeant dans le cluster Kafka.
Si vous devez spécifier plusieurs adresses, séparez-les à l’aide d’une virgule (,).

**Broker list**

Saisissez les adresses des nœuds du broker du cluster Kafka à utiliser.
L’adresse doit se présenter sous la forme suivante : *hostname:port*. Ces informations contiennent le nom et le port du nœud hébergeant dans le cluster Kafka.
Si vous devez spécifier plusieurs adresses, séparez-les à l’aide d’une virgule (,)

**Use SSL/TLS**

Cochez cette case pour activer la connexion chiffrée SSL ou TLS.
Utilisez le composant tSetKeystore dans le même Job afin de spécifier les informations de chiffrement.
Pour plus d’informations concernant le tSetKeystore, consultez tSetKeystore à la page 3745.
Cette case est disponible depuis Kafka 0.9.0.1.

**Use Kerberos authentication**

Si le cluster Kafka à utiliser est sécurisé par Kerberos, cochez cette case pour afficher les paramètres associés à définir :

- **JAAS configuration path** : saisissez le chemin d’accès ou parcourrez votre système jusqu’au fichier de configuration JAAS à utiliser par le Job pour authentification en tant que client à Kafka.
Le fichier JAAS décrit comment les clients, les Jobs Kafka en termes de Talend peuvent se connecter aux nœuds du broker Kafka, en utilisant soit le mode kinit, soit le mode keytab. Il doit être stocké sur la machine où sont exécutés les Jobs.

Talend, Kerberos ou Kafka ne fournissent pas ce fichier JAAS. Vous devez le créer en suivant les explications dans Configuring Kafka client (en anglais), selon la stratégie de sécurité de votre entreprise.

- **Kafka brokers principal name**: saisissez le membre primaire du Principal Kerberos défini pour les brokers lorsque vous avez créé le cluster de brokers. Par exemple, dans ce Principal `kafka/kafka1.hostname.com@EXAMPLE.COM`, le membre primaire à utiliser pour renseigner ce champ est `kafka`.

- **Set kinit command path**: Kerberos utilise un chemin par défaut pour son exécutable kinit. Si vous avez modifié ce chemin, cochez cette case et saisissez votre chemin d'accès personnalisé.

  Si vous laissez cette case décochée, le chemin par défaut est utilisé.

- **Set Kerberos configuration path**: Kerberos utilise un chemin par défaut vers son fichier de configuration, le fichier `krb5.conf` (ou `krb5.ini` sous Windows) pour Kerberos 5 par exemple. Si vous avez modifié ce chemin, cochez cette case et saisissez le chemin d'accès personnalisé au fichier de configuration Kerberos.

  Si vous laissez cette case décochée, une stratégie donnée est appliquée par Kerberos pour tenter de trouver les informations de configuration nécessaires. Pour plus d'informations concernant cette stratégie, consultez la section Locating the krb5.conf Configuration File dans Kerberos requirements (en anglais).

  Pour plus d'informations concernant la manière dont est sécurisé un cluster Kafka via Kerberos, consultez Authenticating using SASL (en anglais).

  Cette case est disponible depuis Kafka 0.9.0.1.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour rassembler les données de log au niveau du Job, ainsi qu’au niveau de chaque composant. |

### Utilisation

| Règle d’utilisation | Ce composant est utilisé en standalone pour créer une connexion à Kafka, que les autres composants Kafka peuvent réutiliser. |
**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tKafkaCreateTopic

Le tKafkaCreateTopic crée un sujet Kafka que les autres composant Kafka peuvent utiliser.

Ce composant vous permet d'exécuter graphiquement une commande afin de créer un sujet avec différentes propriétés au niveau du sujet.

**Propriétés du tKafkaCreateTopic Standard**

Ces propriétés sont utilisées pour configurer le tKafkaCreateTopic s'exécutant dans le framework de Jobs Standard.

Le composant tKafkaCreateTopic Standard appartient à la famille Internet.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>Sélectionnez la version du cluster Kafka à utiliser.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Zookeeper quorum list</strong></td>
<td>Saisissez l'adresse du service Zookeeper pour le cluster Kafka à utiliser.</td>
</tr>
<tr>
<td></td>
<td>L'adresse doit se présenter sous la forme suivante : hostname:port. Ces informations contiennent le nom et le port du nœud hébergeant dans le cluster Kafka.</td>
</tr>
<tr>
<td></td>
<td>Si vous devez spécifier plusieurs adresses, séparez-les à l'aide d'une virgule (,).</td>
</tr>
<tr>
<td><strong>Action on topic</strong></td>
<td>Sélectionnez comment créer un sujet :</td>
</tr>
<tr>
<td></td>
<td>• Create topic : cela crée un sujet. Si le sujet existe déjà, le Job entier est arrêté.</td>
</tr>
<tr>
<td></td>
<td>• Create topic if not exists : cela crée un sujet lorsque le sujet n'existe pas. Si le sujet existe, le Job ignore la création et passe à l'étape suivante.</td>
</tr>
<tr>
<td><strong>Topic name</strong></td>
<td>Saisissez le nom du topic à créer.</td>
</tr>
<tr>
<td><strong>Replication factor</strong></td>
<td>Saisissez le nombre de répliques à créer pour les logs des (l'ensemble des messages) partitions du topic.</td>
</tr>
<tr>
<td><strong>Number of partitions</strong></td>
<td>Saisissez le nombre de partitions à créer pour le topic.</td>
</tr>
<tr>
<td><strong>Set topic retention time (ms)</strong></td>
<td>Cochez cette case pour définir le temps maximal en millisecondes durant lequel le contenu du topic est retenu.</td>
</tr>
</tbody>
</table>
Lorsque le temps est écoule, le contenu est supprimé ou compacté, selon la politique définie dans la table **Topic properties**, dans la vue **Advanced settings**.

**Advanced settings**

<table>
<thead>
<tr>
<th>Topic properties</th>
<th>Ajoutez les propriétés Kafka relatives au topic nécessaires pour personnaliser cette table. Pour plus d’informations concernant les propriétés de niveau topic pouvant être définies dans cette table, consultez la section décrivant la configuration au niveau des topics dans la documentation Kafka à l’adresse <a href="http://kafka.apache.org/documentation.html#topic-config">http://kafka.apache.org/documentation.html#topic-config</a> (en anglais).</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement au niveau du Job ainsi qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est utilisé en standalone pour créer un sujet pouvant être utilisé par les autres composants Kafka, dans d’autres Jobs ou sous-jobs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tKafkaInput

Ce composant transmet des messages à traiter aux composants qui suivent dans le Job que vous créez.

Le tKafkaInput est un broker de messages générique transmettant des messages au Job exécutant les transformations sur ces messages.

Propriétés du tKafkaInput Standard

Ces propriétés sont utilisées pour configurer le tKafkaInput s’exécutant dans le framework de Jobs Standard.

Le composant tKafkaInput Standard appartient à la famille Internet.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Notez que le schéma de ce composant est en lecture seule. Il stocke les messages envoyés du producteur de messages. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Output type</td>
<td>Sélectionnez le type de données à envoyer au composant suivant. De manière générale, il est recommandé d’utiliser des types String, car le tKafkaInput peut traduire automatiquement les messages Kafka byte[] en chaînes de caractères à traiter par le Job. Cependant, si ce format de messages Kafka n’est pas connu par le tKafkaInput, comme le Protobuf, vous pouvez sélectionner byte[] et utiliser un composant Custom code comme le tJavaRow afin de désérialiser les messages en chaînes de caractères afin que les autres composants du même Job puissent traiter ces messages.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Version</td>
<td>Sélectionnez la version du cluster Kafka à utiliser.</td>
</tr>
<tr>
<td><strong>KafkaInput</strong></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Broker list</strong></td>
<td>Si vous devez spécifier plusieurs adresses, séparez-les à l’aide d’une virgule (,). Ce champ est disponible uniquement pour Kafka 0.8.2.0.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Saisissez les adresses des nœuds du broker du cluster Kafka à utiliser.</strong></td>
<td></td>
</tr>
<tr>
<td>L’adresse doit se présenter sous la forme suivante : hostname:port. Ces informations contiennent le nom et le port du nœud hébergeant dans le cluster Kafka.</td>
<td></td>
</tr>
<tr>
<td>Si vous devez spécifier plusieurs adresses, séparez-les à l’aide d’une virgule (,). Ce champ est disponible depuis Kafka 0.9.0.1.</td>
<td></td>
</tr>
<tr>
<td><strong>Reset offsets on consumer group</strong></td>
<td>Cochez cette case pour supprimer les offsets sauvegardés pour le groupe de consommateurs à utiliser, afin que ce groupe de consommateurs soit géré comme un nouveau groupe n’ayant consommé aucun message.</td>
</tr>
<tr>
<td><strong>New consumer group starts from</strong></td>
<td>Sélectionnez le point de départ duquel les messages d’un topic sont consommés.</td>
</tr>
<tr>
<td>Dans Kafka, le numéro d’ID augmentant d’un message se nomme offset. Lorsqu’un nouveau groupe de consommateurs démarre, dans cette liste, vous pouvez sélectionner beginning pour commencer la consommation depuis le message le plus ancien du topic entier ou sélectionner latest pour attender un nouveau message.</td>
<td></td>
</tr>
<tr>
<td>Notez que le groupe de consommateurs prend en compte uniquement les messages dont l’offset a été commité comme point de départ.</td>
<td></td>
</tr>
<tr>
<td>Chaque groupe de consommateurs possède son propre compteur pour se rappeler la position d’un message consommé. Pour cette raison, une fois qu’un groupe de consommateurs à commencé à consommer des messages d’un topic donné, un groupe de consommateurs reconnaît le message le plus récent en voyant simplement la position où son groupe a arrêté la consommation, plutôt que le topic complet. Partant de ce principe, les comportements suivants peuvent être attendus :</td>
<td></td>
</tr>
<tr>
<td>• Si vous reprenez un groupe de consommateurs existant, cette option détermine le point de départ de ce groupe de consommateurs uniquement s’il n’a pas déjà de point de départ commité. Sinon, ce groupe de consommateurs démarre du point de départ commité. Par exemple, un topic contient 100 messages. Si un groupe de consommateurs existant a traité 50 messages et a commité leurs offsets, le même groupe de consommateurs reprend à partir de l’offset 51.</td>
<td></td>
</tr>
<tr>
<td>• si vous créez un nouveau groupe de consommateurs ou en réinitialisez un existant, ce qui signifie que ce groupe n’a consommé aucun message de ce topic,</td>
<td></td>
</tr>
</tbody>
</table>
lorsque vous le démarrez depuis le dernier message, ce nouveau groupe démarre et attend l’offset 101.

<table>
<thead>
<tr>
<th>Offset storage</th>
<th>Sélectionnez le système dans lequel vous souhaitez commiter les offsets des messages consommés.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable dual commit</td>
<td>Si vous sélectionnez <strong>Kafka</strong> comme système de stockage des offsets, la case <em>Enable dual commit</em> est disponible. Elle est cochée par défaut pour permettre au Job de commiter les messages dans Zookeeper et Kafka. Si vous souhaitez que le Job commite uniquement dans Kafka, décochez cette case.</td>
</tr>
</tbody>
</table>
| Auto-commit offsets | Cochez cette case afin de permettre au **tKafkaInput** de sauvegarder automatiquement son état de consommation à la fin de chaque intervalle de temps donné. Vous devez définir cet intervalle dans le champ *Interval* affiché.  
Notez que les offsets sont commités uniquement à la fin de chaque intervalle. Si votre Job s’arrête au milieu d’un intervalle, l’état de consommation du message dans cet intervalle n’est pas commité. |
| Topic name | Saisissez le nom du topic depuis lequel le **tKafkaInput** reçoit le flux de messages. |
| Consumer group ID | Saisissez le nom du groupe de consommateurs auquel vous souhaitez que le consommateur courant (le **tKafkaInput**) appartienne.  
Ce groupe de consommateurs sera créé lors de l’exécution s’il n’existe pas. |
| Stop after a maximum total duration (ms) | Cochez cette case et, dans le champ qui s’affiche, saisissez la durée (en millisecondes) à la fin de laquelle le **tKafkaInput** arrête de s’exécuter. |
| Stop after receiving a maximum number of messages | Cochez cette case et, dans le champ qui s’affiche, saisissez le nombre maximal de messages que vous souhaitez que le **tKafkaInput** reçoive, avant d’arrêter automatiquement son exécution. |
| Stop after maximum time waiting between messages (ms) | Cochez cette case et, dans le champ qui s’affiche, saisissez le temps (en millisecondes) durant lequel le **tKafkaInput** attend un nouveau message. Si le **tKafkaInput** ne reçoit pas de nouveau message lorsque ce temps d’attente est écoulé, il arrête de s’exécuter. |
| Use SSL/TLS | Cochez cette case pour activer la connexion chiffrée SSL ou TLS.  
Utilisez le composant **tSetKeystore** dans le même Job afin de spécifier les informations de chiffrement.  
Pour plus d’informations concernant le **tSetKeystore**, consultez **tSetKeystore** à la page 3745.  
Cette case est disponible depuis Kafka 0.9.0.1. |
### Use Kerberos authentication

Si le cluster Kafka à utiliser est sécurisé par Kerberos, cochez cette case pour afficher les paramètres associés à définir :

- **JAAS configuration path** : saisissez le chemin d’accès ou parcourez votre système jusqu’au fichier de configuration JAAS à utiliser par le Job pour authentification en tant que client à Kafka.

  Le fichier JAAS décrit comment les clients, les Jobs Kafka en termes **Talend** peuvent se connecter aux nœuds du broker Kafka, en utilisant soit le mode kinit, soit le mode keytab. Il doit être stocké sur la machine où sont exécutés les Jobs. **Talend**, Kerberos ou Kafka ne fournissent pas ce fichier JAAS. Vous devez le créer en suivant les explications dans Configuring Kafka client (en anglais), selon la stratégie de sécurité de votre entreprise.

- **Kafka brokers principal name** : saisissez le membre primaire du Principal Kerberos défini pour les brokers lorsque vous avez créé le cluster de brokers. Par exemple, dans ce Principal `kafka/kafka1.hostname.com@EXAMPLE.COM`, le membre primaire à utiliser pour renseigner ce champ est `kafka`.

- **Set kinit command path** : Kerberos utilise un chemin par défaut pour son exécutable kinit. Si vous avez modifié ce chemin, cochez cette case et saisissez votre chemin d’accès personnalisé.

  Si vous laissez cette case décochée, le chemin par défaut est utilisé.

- **Set Kerberos configuration path** : Kerberos utilise un chemin par défaut vers son fichier de configuration, le fichier `krb5.conf` (ou `krb5.ini` sous Windows) pour Kerberos 5 par exemple. Si vous avez modifié ce chemin, cochez cette case et saisissez le chemin d’accès personnalisé au fichier de configuration Kerberos.

  Si vous laissez cette case décochée, une stratégie donnée est appliquée par Kerberos pour tenter de trouver les informations de configuration nécessaires. Pour plus d’informations concernant cette stratégie, consultez la section Locating the `krb5.conf` Configuration File dans Kerberos requirements (en anglais).

  Pour plus d’informations concernant la manière dont est sécurisé un cluster Kafka via Kerberos, consultez Authenticating using SASL (en anglais).

  Cette case est disponible depuis Kafka 0.9.0.1.

### Advanced settings

<table>
<thead>
<tr>
<th>Kafka properties</th>
<th>Ajoutez les propriétés de consommation Kafka nécessaires pour personnaliser cette table. Par exemple, configurez une valeur spécifique <code>zookeeper.conn</code></th>
</tr>
</thead>
</table>
**Téléchargement des offsets avec le message**

Cochez cette case pour écrire en sortie les offsets des messages consommés au composant suivant. Lorsque vous cochez cette case, une colonne *offset en lecture* seule est ajoutée au schéma.

**Custom encoding**

Il est possible de rencontrer des problèmes d'encodage lorsque vous traitez les données stockées. Dans ce cas, cochez cette case pour afficher la liste *Encoding*. Sélectionnez l'encodage à partir de la liste ou sélectionnez *Custom* et définissez-le manuellement.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement au niveau du Job ainsi qu'au niveau de chaque composant.

### Utilisation

**Règle d'utilisation**

Ce composant est utilisé en tant que composant d'entrée et nécessite un lien de sortie. Lorsque le sujet Kafka à utiliser n'existe pas, il peut être utilisé avec le composant *tKafkaCreateTopic* pour lire le sujet créé par ce dernier.

**Scénario associé**

Aucun scénario n'est disponible pour la version Standard de ce composant.
**tKafkaOutput**

Ce composant publie des messages dans un système Kafka.

Ce composant reçoit des messages sérialisés en tableaux d’octets via le composant précédent et écrit ces messages dans un système Kafka donné.

**Propriétés du tKafkaOutput Standard**

Ces propriétés sont utilisées pour configurer le tKafkaOutput s’exécutant dans le framework de Jobs Standard.

Le composant tKafkaOutput Standard appartient à la famille Internet.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Domain</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schema et Edit schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <code>line</code> lors du nommage des champs. Notez que le schéma de ce composant est en lecture seule. Il stocke les messages à publier.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>Sélectionnez la version du cluster Kafka à utiliser.</td>
</tr>
<tr>
<td><strong>Broker list</strong></td>
<td>Saisissez les adresses des nœuds du broker du cluster Kafka à utiliser.</td>
</tr>
<tr>
<td></td>
<td>L’adresse doit se présenter sous la forme suivante : <code>hostname:port</code>. Ces informations contiennent le nom et le port du nœud hébergeant dans le cluster Kafka.</td>
</tr>
<tr>
<td></td>
<td>Si vous devez spécifier plusieurs adresses, séparez-les à l’aide d’une virgule (,).</td>
</tr>
<tr>
<td><strong>Topic name</strong></td>
<td>Saisissez le nom du topic dans lequel vous souhaitez publier des messages. Ce topic doit déjà exister.</td>
</tr>
<tr>
<td><strong>Compress the data</strong></td>
<td>Cochez la case Compress the data afin de compresser les données de sortie.</td>
</tr>
<tr>
<td><strong>Use SSL/TLS</strong></td>
<td>Cochez cette case pour activer la connexion chiffrée SSL ou TLS.</td>
</tr>
<tr>
<td></td>
<td>Utilisez le composant tSetKeystore dans le même Job afin de spécifier les informations de chiffrement.</td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant le **tSetKeystore**, consultez [tSetKeystore](#) à la page 3745. 
Cette case est disponible depuis Kafka 0.9.0.1.

### Use Kerberos authentication

Si le cluster Kafka à utiliser est sécurisé par Kerberos, cochez cette case pour afficher les paramètres associés à définir :

- **JAAS configuration path** : saisissez le chemin d’accès ou parcourez votre système jusqu’au fichier de configuration JAAS à utiliser par le Job pour authentification en tant que client à Kafka.

Le fichier JAAS décrit comment les clients, les Jobs Kafka en termes *Talend* peuvent se connecter aux nœuds du broker Kafka, en utilisant soit le mode kinit, soit le mode keytab. Il doit être stocké sur la machine où sont exécutés les Jobs. *Talend*, Kerberos ou Kafka ne fournissent pas ce fichier JAAS. Vous devez le créer en suivant les explications dans Configuring Kafka client (en anglais), selon la stratégie de sécurité de votre entreprise.

- **Kafka brokers principal name** : saisissez le membre primaire du Principal Kerberos défini pour les brokers lorsque vous avez créé le cluster de brokers. Par exemple, dans ce Principal *kafka/kafka1.hostname.com@EXAMPLE.COM*, le membre primaire à utiliser pour renseigner ce champ est *kafka*.

- **Set kinit command path** : Kerberos utilise un chemin par défaut pour son exécutable kinit. Si vous avez modifié ce chemin, cochez cette case et saisissez votre chemin d’accès personnalisé.

Si vous laissez cette case décochée, le chemin par défaut est utilisé.

- **Set Kerberos configuration path** : Kerberos utilise un chemin par défaut vers son fichier de configuration, le fichier *krb5.conf* (ou *krb5.ini* sous Windows) pour Kerberos 5 par exemple. Si vous avez modifié ce chemin, cochez cette case et saisissez le chemin d’accès personnalisé au fichier de configuration Kerberos.

Si vous laissez cette case décochée, une stratégie donnée est appliquée par Kerberos pour tenter de trouver les informations de configuration nécessaires. Pour plus d’informations concernant cette stratégie, consultez la section Locating the *krb5.conf* Configuration File dans Kerberos requirements (en anglais).

Pour plus d’informations concernant la manière dont est sécurisé un cluster Kafka via Kerberos, consultez [Authenticating using SASL](#) (en anglais).

Cette case est disponible depuis Kafka 0.9.0.1.
**Advanced settings**

| **Kafka properties**                          | Ajoutez dans cette table les nouvelles propriétés producteur Kafka à personnaliser.
|                                               | Pour plus d’informations concernant les nouvelles propriétés producteur que vous pouvez définir dans cette table, consultez la section décrivant la nouvelle configuration producteur dans la documentation Kafka à l’adresse [http://kafka.apache.org/documentation.html #newproducerconfigs](http://kafka.apache.org/documentation.html #newproducerconfigs) (en anglais). |

| **tStatCatcher Statistics**                   | Cochez cette case pour collecter les métadonnées de traitement au niveau du Job ainsi qu’au niveau de chaque composant. |

**Utilisation**

| **Règle d’utilisation**                        | Ce composant est un composant de fin. Il nécessite un composant `tJavaRow` ou `tJava` pour transformer les données entrantes en tableaux d’octets sérialisés.
|                                               | L’exemple suivant vous montre comment construire une instruction pour effectuer cette transformation :

```java
output_row.serializedValue = input_row.users.getBytes();
```

|                                               | Dans ce code, la variable `output_row` représente le schéma des données à écrire en sortie dans le `tKafkaOutput` et `output_row.serializedValue` la colonne en lecture seule de ce schéma. La variable `input_row` représente le schéma des données entrantes et `input_row.users` la colonne d’entrée nommée `users` à transformer en tableau d’octets via la méthode `getBytes()`. |

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tLDAPAttributesInput

Ce composant analyse chaque élément trouvé via une requête LDAP et liste une série d’attributs associés à l’objet.

Le tLDAPAttributesInput exécute une LDAP basée sur un filtre prédéfini et correspondant au schéma défini. Puis la liste de champ est transmise au composant suivant via une connexion Main row.

Propriétés du tLDAPAttributesInput Standard

Ces propriétés sont utilisées pour configurer le tLDAPAttributesInput s’exécutant dans le framework de Jobs Standard.

Le composant tLDAPAttributesInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Use an existing connection

Cochez cette case et cliquez sur le composant tLDAPConnection dans la liste Component List, afin de réutiliser les informations de la connexion que vous avez précédemment définie.

Host
Adresse IP du serveur d’annuaire LDAP.

Port
Numéro du port d’écoute du serveur.

Base DN
Chemin d’accès à l’arborescence de l’utilisateur autorisé.

Protocol
 Sélectionnez le type de protocole dans la liste.

LDAP : aucun codage n’est utilisé.

LDAPS : LDAP sécurisé. Quand cette option est sélectionnée, la case Advanced CA apparaît. Une fois sélectionné, le mode avancé vous permet de spécifier le répertoire et le mot de passe de la clé secrète du certificat pour le stockage d’un CA spécifique. Cependant, vous pouvez toujours désactiver la validation du certificat, en cochant la case Trust all certs.

TLS : le certificat est utilisé. Quand cette option est sélectionnée, la case Advanced CA apparaît et est utilisée de la même façon que pour le type LPDAPS.

Authentication User et Password
Cochez Authentication si une connexion LDAP est nécessaire. Notez que le login doit correspondre
à la syntaxe LDAP pour être valide. Par exemple :
"cn=Directory Manager".

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

Filter
Saisissez le filtre comme attendu par l’annuaire LDAP db.

Multi valued field separator
Saisissez le séparateur de valeur des champs à valeurs multiples.

Alias dereferencing
Sélectionnez l’option dans la liste. Never permet l’amélioration des performances de recherche si vous êtes sûr qu’aucun alias n’est déréférencé. Par défaut, utilisez Always :
- Always : les alias sont toujours déréférencés.
- Never : les alias ne sont jamais déréférencés.
- Searching : cette option déréférence les alias uniquement après la résolution du nom.
- Finding : cette option déréférence les alias uniquement lors de la résolution du nom.

Referral handling
Sélectionnez l’option dans la liste :
- Ignore : ne tient pas compte des redirections des requêtes
- Follow : tient compte des redirections des requêtes

Limit
Cette option permet de limiter le nombre d’enregistrements lus, si nécessaire.

Time Limit
Cette option permet de limiter la durée de connexion à l’annuaire.

Paging
Spécifiez le nombre d’entrées retournées en une fois par le serveur LDAP.

Die on error
Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject.

Schema et Edit Schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.


**Avertissement** :
Comme ce composant sert à lister les attributs associés à un élément LDAP, son schéma est prédéfini. Conservez ces colonnes définies, même si vous avez besoin d’ajouter de nouvelles colonnes. Pour cela, utilisez le mode Built-In.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

Le schéma prédéfini comprend :
- **objectclass** : liste des classes d’éléments
- **mandatoryattributes** : liste des attributs obligatoires pour ces classes
- **optionalattributes** : liste des attributs facultatifs pour ces classes
- **objectattributes** : liste des attributs essentiels aux éléments analysés.

**Advanced settings**

<table>
<thead>
<tr>
<th>Class Definition Root</th>
<th>Spécifiez la racine de l’espace de nom de l’élément Class definition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| Global Variables | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **RESULT_NAME** : nom de l’entrée LDAP correspondant au filtre de recherche. Cette variable est une variable **Flow** et retourne une chaîne de caractères. |
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | Ce composant couvre toutes les requêtes LDAP possibles. Note : Appuyez sur *Ctrl + Espace* pour accéder à la liste des variables globales, notamment la variable *GetResultName* permettant de récupérer automatiquement la base correspondante. |

### Scénario associé

Le composant *tLDAPAttributesInput* fonctionne de manière similaire au *tLDAPInput*. Vous pouvez donc consulter Scénario : Afficher le contenu filtré d’un annuaire LDAP à la page 2070.
tLDAPClose

Ce composant ferme une connexion à un serveur d’un annuaire LDAP afin de libérer des ressources occupées.

**Propriétés du tLDAPClose Standard**

Ces propriétés sont utilisées pour configurer le tLDAPClose s’exécutant dans le framework de Jobs Standard.

Le composant tLDAPClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Component list | S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tLDAPConnection dans la liste. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau des composants. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec des composants LDAP, notamment avec le tLDAPConnection. |

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les...
Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tLDAPConnection

Ce composant crée une connexion au serveur d’un annuaire LDAP.
Cette connexion peut être invoquée par d’autres composants devant accéder à l’annuaire LDAP, c’est-à-dire les composants tLDAPInput, tLDAPOutput, etc.

Propriétés du tLDAPConnection Standard

Ces propriétés sont utilisées pour configurer le tLDAPConnection s’exécutant dans le framework de Jobs Standard.
Le composant tLDAPConnection Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur d’annuaire LDAP.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Protocol</td>
<td>Sélectionnez le type de protocole dans la liste.</td>
</tr>
<tr>
<td></td>
<td>LDAP : aucun codage n’est utilisé</td>
</tr>
<tr>
<td></td>
<td>LDAPS : LDAP sécurisé. Quand cette option est sélectionnée, la case Advanced CA apparaît. Une fois sélectionné, le mode avancé vous permet de spécifier le répertoire et le mot de passe de la clé secrète du certificat pour le stockage d’un CA spécifique. Cependant, vous pouvez toujours désactiver la validation du certificat, en cochant la case Trust all certs.</td>
</tr>
<tr>
<td></td>
<td>TLS : le certificat est utilisé. Quand cette option est sélectionnée, la case Advanced CA apparaît et est utilisée de la même façon que pour le type LPDAPS.</td>
</tr>
<tr>
<td>Base DN</td>
<td>Chemin d’accès à l’arborescence de l’utilisateur autorisé.</td>
</tr>
<tr>
<td>User et Password</td>
<td>Renseignez les champs User et Password comme requis par l’annuaire. Notez que le login doit correspondre à la syntaxe LDAP pour être valide. Par exemple : &quot;cn=Directory Manager&quot;.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets</td>
</tr>
</tbody>
</table>
doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

### Alias dereferencing

Sélectionnez l’option dans la liste. **Never** permet l’amélioration des performances de recherche si vous êtes sûr qu’aucun alias n’est déréférencé. Par défaut, utilisez **Always** :

- **Always** : les alias sont toujours déréférencés.
- **Never** : les alias ne sont jamais déréférencés.
- **Searching** : cette option déréférence les alias uniquement après la résolution du nom.
- **Finding** : cette option déréférence les alias uniquement lors de la résolution du nom.

### Referral handling

Sélectionnez l’option dans la liste :

- **Ignore** : ne tient pas compte des redirections des requêtes.
- **Follow** : tient compte des redirections des requêtes.

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant doit être utilisé avec d’autres composants LDAP, notamment les composants <strong>tLDAPInput</strong> et <strong>tLDAPOutput</strong>.</th>
</tr>
</thead>
</table>

### Scénario associé

Ce composant est étroitement lié au **tLDAPInput** et au **tLDAPOutput**, puisqu’il vous permet de ne pas avoir à toujours saisir les informations de connexion lors de l’utilisation de différents **tLDAPInput** ou **tLDAPOutput**.

Pour un scénario associé au **tLDAPConnection**, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tLDAPInput

Ce composant exécute une requête LDAP basée sur un filtre prédéfini et correspondant au schéma défini. Puis la liste de champ est transmise au composant suivant via une connexion Main row.

Propriétés du tLDAPInput Standard

Ces propriétés sont utilisées pour configurer le tLDAPInput s’exécutant dans le framework de Jobs Standard.
Le composant tLDAPInput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres de connexion que vous avez déjà définie. |
|----------------------------| Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |

| Host | Adresse IP du serveur d’annuaire LDAP. |
| Port | Numéro du port d’écoute du serveur. |
| Base DN | Chemin d’accès à l’arborescence de l’utilisateur autorisé. **Remarque :** Pour retrouver les informations complètes du Distinguished Name, ajoutez un champ appelé DN dans le schéma, en minuscules ou en majuscules. |
| Protocol | Sélectionnez le type de protocole dans la liste. |
**LDAP :** aucun codage n'est utilisé

**LDAPS :** LDAP sécurisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît. Une fois sélectionné, le mode avancé vous permet de spécifier le répertoire et le mot de de passe de la clé secrète du certificat pour le stockage d'un CA spécifique. Cependant, vous pouvez toujours désactiver la validation du certificat, en cochant la case **Trust all certs**.

**TLS :** le certificat est utilisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît et est utilisée de la même façon que pour le type **LDAPS**.

| Authentication User et Password | Cochez **Authentication** si une connexion LDAP est nécessaire. Notez que le login doit correspondre à la syntaxe LDAP pour être valide. Par exemple : "cn=Directory Manager".

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Saisissez le filtre comme attendu par l'annuaire LDAP db.</td>
</tr>
<tr>
<td>Multi valued field separator</td>
<td>Saisissez le séparateur de valeur des champs à valeurs multiples.</td>
</tr>
</tbody>
</table>
| Alias dereferencing | Sélectionnez l’option dans la liste. **Never** permet l’amélioration des performances de recherche si vous êtes sûr qu’aucun alias n’est déréférencé. Par défaut, utilisez **Always** :

- **Always** : les alias sont toujours déréférencés.
- **Never** : les alias ne sont jamais déréférencés.
- **Searching** : cette option déréférence les alias uniquement après la résolution du nom.
- **Finding** : cette option déréférence les alias uniquement lors de la résolution du nom. |
| Referral handling | Sélectionnez l’option dans la liste :

- **Ignore** : ne tient pas compte des redirections des requêtes.
- **Follow** : tient compte des redirections des requêtes. |
| Limit | Cette option permet de limiter le nombre d’enregistrements lus, si nécessaire. |
| Time Limit | Cette option permet de limiter la durée de connexion à l’annuaire. |
| Paging | Spécifiez le nombre d’entrées retournées en une fois par le serveur LDAP. |
| Die on error | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le Job. |
traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject.

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Avertissement :**

Seuls trois types de données sont supportés : String, byte[] et List. Vous pouvez utiliser un tMap pour effectuer une conversion, si nécessaire.

**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RESULT_NAME : nom de l’entrée LDAP correspondant au filtre de recherche. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>
**tLDAPInput**

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

**Règle d’utilisation**

Ce composant couvre toutes les requêtes LDAP possibles.

Note : Appuyez sur **Ctrl + Espace** pour accéder à la liste des variables globales, inclus la variable **GetResultName** permettant de récupérer automatiquement la base correspondante.

### Scénario : Afficher le contenu filtré d’un annuaire LDAP

Le Job décrit ci-dessous filtre simplement l’annuaire LDAP et affiche le résultat dans la console de la vue **Run**.

![Diagramme de flux](image)

**Ajouter et relier les composants**

**Procédure**

Cliquez et déposez les composants **tLDAPInput** et **tLogRow**.

### Configurer les composants

**Procédure**

1. Paramétrez les propriétés du **tLDAPInput**.
2. Sélectionnez **Repository** dans le champ **Property type** si vous avez stocké les informations de connexion LDAP dans le **Metadata Manager** du **Repository**. Puis sélectionnez l’entrée adéquate dans la liste.
3. En mode **Built-In**, renseignez manuellement les champs **Host** et **Port**. Host peut être l’adresse IP du serveur de l’annuaire LDAP ou son nom DNS.
4. Aucune **Base DN** particulière ne doit être paramétrée.

6. Cochez la case **Authentication** et renseignez les informations de connexion pour lire l’annuaire, si nécessaire. Dans ce scénario, aucune authentification n’est nécessaire.

7. **In the Filter area**, type in the command, the data selection is based on. In this example, the filter is: `(&(objectClass/inetorgperson)&(uid=PIERRE DUPONT))`. Dans la zone **Filter**, saisissez la commande sur laquelle est basée la sélection de données. Dans cet exemple, le filtre est : `(&(objectClass/inetorgperson)&(uid=PIERRE DUPONT))`.

8. Renseignez le champ **Multi-valued field separator** avec une virgule, puisque certains champs contiennent plus d’une valeur séparée par une virgule.

9. Comme vous ne savez pas si des alias sont utilisés dans l’annuaire LDAP, sélectionnez **Always** dans la liste.

10. Sélectionnez **Ignore** dans le champ **Referral handling**.

11. Dans le champ **Limit**, définissez la limite à **100** dans cet exemple.

12. Définissez le **Schéma** tel qu’il est requis par l’annuaire LDAP. Dans cet exemple, le schéma est composé de six colonnes dont les colonnes **objectClass** et **uid** qui seront filtrées.

<table>
<thead>
<tr>
<th>Column</th>
<th>Db Column</th>
<th>Key</th>
<th>Type</th>
<th>Nullable</th>
<th>Date P...</th>
<th>Length</th>
<th>Pre...</th>
</tr>
</thead>
<tbody>
<tr>
<td>dc</td>
<td>dc</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td>255</td>
<td></td>
</tr>
<tr>
<td>ou</td>
<td>ou</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td>255</td>
<td></td>
</tr>
<tr>
<td>objectClass</td>
<td>objectClass</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td>255</td>
<td></td>
</tr>
<tr>
<td>mail</td>
<td>mail</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td>255</td>
<td></td>
</tr>
<tr>
<td>uid</td>
<td>uid</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td>255</td>
<td></td>
</tr>
<tr>
<td>dn</td>
<td>dn</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td>255</td>
<td></td>
</tr>
</tbody>
</table>

13. Dans le composant **tLogRow**, aucune configuration particulière n’est exigée.
Résultats

|DATA|top.person.organizationalPerson/inetorgperson.s4400user|nhirt78@talend.com|PIERRE DUFONT|
Job testLDAPInput ended at 18:05 18/09/2007. [exit code=0]

Seule une entrée de l'annuaire correspond aux critères du filtre défini dans le composant **tLDAPInput**.
**tLDAPOutput**

Ce composant exécute une requête LDAP basée sur un filtre prédéfini et correspondant à un schéma défini. Puis la liste de champ est transmise au composant suivant via une connexion Main row.

### Propriétés du tLDAPOutput Standard

Ces propriétés sont utilisées pour configurer le tLDAPOutput s’exécutant dans le framework de Jobs Standard.

Le composant tLDAPOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

#### Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.

Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur d’annuaire LDAP.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro de port d’écoute du serveur.</td>
</tr>
<tr>
<td>Base DN</td>
<td>Chemin d’accès à l’arborescence de l’utilisateur autorisé.</td>
</tr>
</tbody>
</table>

**Protocol**

Sélectionnez le type de protocole dans la liste.

- **LDAP** : aucun codage n’est utilisé
- **LDAPS** : LDAP sécurisé. Quand cette option est sélectionnée, la case Advanced CA apparaît. Une fois sélectionné, le mode avancé vous permet de spécifier le répertoire et le mot de passe de la clé secrète du
certificat pour le stockage d'un CA spécifique. Cependant, vous pouvez toujours désactiver la validation du certificat, en cochant la case Trust all certs.

**TLS** : le certificat est utilisé. Quand cette option est sélectionnée, la case Advanced CA apparaît et est utilisée de la même façon que pour le type LPDAPS.

**User et Password**
Renseignez les champs User (utilisateur) et Password (mot de passe) conformément à l’annuaire.

Notez que le login doit correspondre à la syntaxe LDAP pour être valide. Par exemple : “cn=Directory Manager”.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Multi valued field separator**
Saisissez le séparateur de valeurs dans les champs multi-valués.

**Alias dereferencing**
 Sélectionnez l’option dans la liste. Never permet l’amélioration des performances de recherche si vous êtes sûr qu’aucun alias n’est déréférencé. Par défaut, utilisez Always.

- *Always* : les alias sont toujours déréférencés.
- *Never* : les alias ne sont jamais déréférencés.
- *Searching* : cette option déréférence les alias uniquement après la résolution du nom.
- *Finding* : cette option déréférence les alias uniquement lors de la résolution du nom.

**Referral handling**
Sélectionnez l’option dans la liste :
- *Ignore* : ne tient pas compte des redirections des requêtes.
- *Follow* : tient compte des redirections des requêtes.

**Insert mode**
Sélectionnez l’action dans la liste :
- *Add* : ajoute une valeur dans un attribut multi-valué
- *Insert* : insère de nouvelles données
- *Update* : met à jour les données existantes
- *Delete* : supprime les données sélectionnées de l’annuaire
- *Insert or update* : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.

⚠️ **Avertissement** :
Lorsque l’action Update est sélectionnée, allez dans la vue Advanced settings et cochez la case Use Attribute Options (for update mode) pour afficher la table Attribute Options. Sélectionnez l’attribut à mettre à jour dans
<table>
<thead>
<tr>
<th><strong>DN Column Name</strong></th>
<th>Sélectionnez, à partir de la liste déroulante, le type d’entrée LDAP utilisé.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schema et Edit schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <code>line</code> lors du nommage des champs.</td>
</tr>
<tr>
<td><strong>Built-In</strong></td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
| **Edit schema** | Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
  - **View schema** : sélectionnez cette option afin de voir le schéma.  
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content. |
| **Die on error** | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject. |
Advanced settings

<table>
<thead>
<tr>
<th>Use Attribute Options (for update mode)</th>
<th>Cochez cette case afin de sélectionner l’attribut souhaité (y compris dn, dc, ou, objectClass, mail et uid) et l’opération correspondante (notamment Add, Replace, Remove Attribute, et Remove Value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les requêtes LDAP possibles.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Note : Appuyez sur Ctrl+Espace pour accéder à la liste des variables globales, inclus la variable GetResultName permettant de récupérer automatiquement la base correspondante. Ce composant permet de faire des actions sur une table ou les données d’une table d’une base de données. Il permet aussi de créer un flux de rejet avec un lien Row &gt; Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.</td>
</tr>
</tbody>
</table>

Scénario : Editer des données dans un annuaire LDAP

Le scénario suivant décrit un Job qui lit un annuaire LDAP, met à jour l’e-mail d’une entrée sélectionnée et affiche la sortie avant de mettre à jour l’annuaire LDAP en sortie. C’est-à-dire qu’aucun alias n’est déréférencé et qu’aucune requête n’est redirigée. Ce scénario est basé sur le scénario du
LDAPInput (cf. Scénario : Afficher le contenu filtré d’un annuaire LDAP à la page 2070). Le résultat obtenu est une entrée simple, correspondant à une personne organisationnelle dont l’e-mail est mis à jour.

La sortie présente les champs suivants : *dn, uid* et *mail*, comme défini dans le Job.

**Déposer les composants**

**Procédure**

1. Cliquez et déposez les composants **tLDAPInput, tLDAPOutput, tMap** et **tLogRow** dans l’espace de modélisation.
2. Connectez le composant **tLDAPInput** au **tMap** à l’aide d’un lien **Row > Main**.
3. Reliez le **tMap** au **tLogRow** à l’aide d’un lien **Row > Main**.
4. Reliez le **tLogRow** au **tLDAPOutput** à l’aide d’un lien **Row > Main**.

**Configurer les composants**

**Procédure**

1. Dans la vue **Component** du **tLDAPInput**, paramétrez les informations de connexion au serveur d’annuaire LDAP, ainsi que les filtres tel que décrit dans Scénario : Afficher le contenu filtré d’un annuaire LDAP à la page 2070.

Simplifiez le schéma en enlevant les champs suivants : *dc, ou, objectclass*.

2. Ouvrez le mapper pour définir les changements à exécuter.
Glissez et déposez la colonne *uid* de la table d’entrée (input) à la table de sortie (output) puisque aucune modification n’est nécessaire pour cette colonne.

3. Dans le champ **Expression** de la colonne *dn* de la table de sortie (output), saisissez l’expression exacte attendue par le serveur LDAP afin qu’il atteigne l’arborescence souhaitée et permette d’écrire dans l’annuaire, à condition que vous n’ayez pas déjà renseigné le champ **Base DN** du composant **tLDAPOutput**.
4. Dans cet exemple, la variable globale **GetResultName** est utilisée pour récupérer automatiquement le chemin d'accès. Appuyez sur **Ctrl+Espace** pour accéder à la liste de variable et sélectionnez **tLDAPInput_1_RESULT_NAME**.

5. Dans le champ **Expression** de la colonne *mail*, saisissez l'adresse e-mail qui remplacera celle obsolète dans l'annuaire LDAP. Dans cet exemple, le nouvel e-mail est : *Pierre.Dupont@talend.com*. Cliquez sur **OK** pour valider les modifications.

6. Sélectionnez le composant **tLDAPOutput** pour définir les propriétés d'écriture dans l'annuaire.

7. Définissez manuellement les champs **Port** et **Host** si ces informations ne sont pas stockées dans le **Repository**.

8. Dans le champ **Base DN**, définissez la plus haute arborescence à laquelle vous avez accès. Si vous n'avez pas défini le chemin complet et exact au DN cible, définissez-le ici. Dans cet exemple, le DN complet est fourni par la sortie *dn* du composant **tMap**, Ainsi, seule la plus haute arborescence accessible est fournie : *o=directoryRoot*.

9. Sélectionnez le protocole à utiliser : **LDAP** pour cet exemple.

Puis renseignez les champs **User** et **Password** attendus par l'annuaire LDAP.

Dans le champ **Multi-valued field separator**, saisissez une virgule afin de séparer les champs contenant plusieurs valeurs séparées par des virgules.

10. Laissez les paramètres par défaut des champs **Alias Dereferencing** et **Referral Handling**, c'est-à-dire respectivement **Always** et **Ignore**.

Dans le champ **Insert mode**, sélectionnez l’option **Update** pour cet exemple (l’adresse e-mail). Le schéma est récupéré du composant précédent à l'aide d'une opération de propagation.

11. Dans la vue **Advanced settings**, cochez la case **Use Attributes Options (for update mode)** pour afficher la table **Attribute Options**.

Sélectionnez l'attribut *mail* dans la colonne **Attribute Name** et sélectionnez **Replace** dans la colonne **Option**.
Exécuter le Job

Procédure

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Exécutez le Job en appuyant sur la touche **F6**.

Résultats

uid=PIERRE DUPONT,ou=DATA,o=TALEND,c=TALEND|PIERRE DUPONT|Pierre.Dupont@telend.con
Job LDAPInputnew ended at 14:17 20/09/2007. [exit code=0]
tLDAPRenameEntry

Ce composant renomme une ou des entrée(s) dans un annuaire LDAP sélectionné.

Propriétés du tLDAPRenameEntry Standard

Ces propriétés sont utilisées pour configurer le tLDAPRenameEntry s’exécutant dans le framework de Jobs Standard.

Le composant tLDAPRenameEntry Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong> :</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong> :</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Use an existing connection**

Cochez cette case et cliquez sur le composant **tLDAPConnection** dans la liste **Component List**, afin de réutiliser les informations de la connexion que vous avez précédemment définie.

**Host**

Adresse IP du serveur d’annuaire LDAP.

**Port**

Numéro du port d’écoute du serveur.

**Base DN**

Chemin d’accès à l’arborescence de l’utilisateur autorisé.

**Protocol**

 Sélectionnez le type de protocole dans la liste.

**LDAP** : aucun codage n’est utilisé

**LDAPS** : LDAP sécurisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît. Une fois sélectionné, le mode avancé vous permet de spécifier le répertoire et le mot de passe de la clé secrète du certificat pour le stockage d’un CA spécifique. Cependant, vous pouvez toujours désactiver la validation du certificat, en cochant la case **Trust all certs**.

**TLS** : le certificat est utilisé. Quand cette option est sélectionnée, la case **Advanced CA** apparaît et est utilisée de la même façon que pour le type **LPDAPS**.

**User et Password**

Renseignez les informations d’authentification de l’utilisateur. Notez que le login doit correspondre à la syntaxe LDAP pour être valide. Par exemple : “cn=Directory Manager”.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue
qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

### Alias dereferencing

 Sélectionnez l’option dans la liste. **Never** permet l’amélioration des performances de recherche si vous êtes sûr qu’aucun alias n’est déréférencé. Par défaut, utilisez **Always** :

- **Always** : les alias sont toujours déréférencés.
- **Never** : les alias ne sont jamais déréférencés.
- **Searching** : cette option déréférence les alias uniquement après la résolution du nom.
- **Finding** : cette option déréférence les alias uniquement lors de la résolution du nom.

### Referrals handling

 Sélectionnez l’option dans la liste :

- **Ignore** : ne tient pas compte des redirections des requêtes.
- **Follow** : tient compte des redirections des requêtes.

### Previous DN et New DN

 Sélectionnez à partir de la liste déroulante la colonne du schéma portant l’ancien DN (Previous DN) et celle portant le nouveau DN à appliquer (New DN).

### Schema et Edit Schema

 Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le **Repository**.

 Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

- **Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Die on error
Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau des composants.

## Global Variables

| Global Variables | NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.  
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.  
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |
|-----------------|--------------------------------------------------|

## Utilisation

| Règle d’utilisation | Ce composant couvre toutes les requêtes LDAP possibles. Il est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé en association avec d’autres composants.  
Note : Appuyez sur Ctrl + Espace pour accéder à la liste des variables globales, inclus la variable GetResultName permettant de récupérer automatiquement la base correspondante. |
|---------------------|--------------------------------------------------|

## Scénarios associés

Pour un scénario associé au composant tLDAPRenameEntry, consultez :
- Scénario : Afficher le contenu filtré d’un annuaire LDAP à la page 2070 du composant tLDAPInput.
- Scénario : Editer des données dans un annuaire LDAP à la page 2076 du tLDAPOutput.
**tLibraryLoad**

Ce composant charge des librairies Java utilisables dans un Job.

**Propriétés du tLibraryLoad Standard**

Ces propriétés sont utilisées pour configurer le tLibraryLoad s’exécutant dans le framework de Jobs Standard.

Le composant tLibraryLoad Standard appartient à la famille Custom Code.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Library | Sélectionnez dans la liste la librairie à importer, ou cliquez sur le bouton [...] afin de parcourir votre répertoire et choisir la librairie que vous souhaitez importer. |

**Advanced settings**

| Dynamic Libs | Lib Paths : Saisissez entre guillemets le chemin d’accès à votre librairie. |
| Import | Saisissez le code Java permettant d’importer, si nécessaire, la librairie externe utilisée dans le champ d’édition de code de l’onglet Basic settings des composants tels que le tJava et le tJavaFlex. |

**Global Variables**

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en standalone, mais il est plus logique de l’utiliser dans un Job.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Le chargement de votre librairie se fait en local.</td>
</tr>
</tbody>
</table>

**Scénario : Vérifier le format d’une adresse e-mail**

Ce scénario comprend deux composants, un `tLibraryLoad` et un `tJava`. Le but de ce scénario est de vérifier le format d’une adresse e-mail et de retourner si elle est valide ou non.

**Construire le Job**

**Procédure**

1. Dans la Palette, ouvrez le dossier Custom Code, puis glissez-déposez les composants `tLibraryLoad` et `tJava` dans l’espace de modélisation graphique.
2. Reliez le `tLibraryLoad` au `tJava` à l’aide d’un lien `Trigger > OnSubjobOk`.

**Configurer le composant `tLibraryLoad`**

**Procédure**

1. Double-cliquez sur le `tLibraryLoad` pour afficher sa vue Basic settings. Dans la liste Library, sélectionnez jakarta-oro-2.0.8.jar.
2. Dans le champ Import de l’onglet Advanced settings, saisissez import org.apache.orotext.regex.*;
Configurer le composant tJava

Procédure

1. Double-cliquez sur le composant tJava pour afficher sa vue Component.
2. Dans l'onglet des Basic settings, saisissez votre code, comme dans la capture d'écran. Ce code permet de vérifier si une chaîne de caractères correspond à une adresse e-mail, en se basant sur l'expression régulière suivante : "^[\w_.-]+@[\w_.-]+\.[\w]+$".

```java
//import java.util.List;
import org.apache.oro.text.regex.*;

String email="ma@com.com";

//Validate if the email string is correct or not based on a regex pattern

Perl5Matcher matcher = new Perl5Matcher();
Perl5Compiler compiler = new Perl5Compiler();
Pattern pattern = compiler.compile("^[\w_.-]+@[\w_.-]+\.[\w]+$");

if (!matcher.matches(email, pattern)) {
    System.out.println(email + " : " + false);
} else
    System.out.println(email + " : " + true);
```

Exécuter le Job

Appuyez sur F6 afin d'enregistrer et d'exécuter votre Job.
La **Console** affiche *false*, c'est-à-dire que l'adresse e-mail n'est pas valide. En effet, son format n'est pas correct.
tLineChart

Ce composant lit des données à partir d'un flux d'entrée et transforme les données en graphique en lignes dans un fichier image au format PNG afin de faciliter l'analyse technique.

Propriétés du tLineChart Standard

Ces propriétés sont utilisées pour configurer le tLineChart s'exécutant dans le framework de Jobs Standard.
Le composant tLineChart Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Remarque :
Le schéma du tLineChart contient trois colonnes en lecture seule nommées respectivement series (type String), category (type String), et value (type Integer), selon un ordre prédéterminé. Les données de toutes les autres colonnes seront simplement passées au composant suivant, s'il y en a un, mais elles ne seront pas présentes dans le diagramme en barres.

| Built-in | Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


| Sync columns | Permet de synchroniser le schéma du fichier d’entrée avec le schéma du fichier de sortie. La fonction Sync ne s’affiche que lorsqu’une connexion de type Row est connectée au composant de sortie.

| Generated image path | Nom et chemin d'accès du fichier image de sortie.

| Chart title | Saisissez le titre du graphique en lignes à générer.

| Domain axis label et Range axis label | Saisissez le nom de l’axe domain (axe X) et le nom de l’axe range (axe Y).

<p>| Plot orientation | Choisissez l'orientation qu’aura votre graphique en lignes : VERTICAL ou HORIZONTAL. |</p>
<table>
<thead>
<tr>
<th>Include legend</th>
<th>Saisissez true dans ce champ si vous souhaitez inclure une légende indiquant les lignes des séries dans des couleurs différentes ; sinon, saisissez false.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image width et Image height</td>
<td>Saisissez la largeur et la hauteur du fichier image, exprimées en pixels.</td>
</tr>
<tr>
<td>Moving average</td>
<td>Cochez cette case afin d’afficher la moyenne mobile de chaque série dans votre graphique en lignes. Lorsque cette case est cochée, le champ Period apparaît et vous pouvez définir la période dans laquelle apparaît la moyenne mobile.</td>
</tr>
<tr>
<td>Lower bound and Upper bound</td>
<td>Définissez quelles sont les plus hautes et les plus basses valeurs qui seront affichées sur l’axe range.</td>
</tr>
<tr>
<td>Chart background et Plot background</td>
<td>Choisissez dans la liste la couleur de fond du graphique ainsi que la couleur de fond du plan du graphique (zone de l’orientation).</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant est un composant de sortie. Par conséquent, il requiert un composant d’entrée ainsi qu’une connexion de type Row Main. |
Scénario : Créer un graphique en lignes afin de faciliter l’analyse des tendances

Ce scénario décrit un Job simple qui lit les données d’un fichier CSV et les transforme en graphique en lignes afin de faciliter l’analyse des tendances. Le fichier d’entrée indique combien de temps (en minutes) par semaine une personne regarde différentes chaînes de télévision, sur une période de dix semaines, comme présenté ci-dessous :

<table>
<thead>
<tr>
<th>Week; TV_A; TV_B; TV_C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1; 327; 286; 244</td>
</tr>
<tr>
<td>2; 326; 285; 243</td>
</tr>
<tr>
<td>3; 325; 283; 245</td>
</tr>
<tr>
<td>4; 323; 282; 246</td>
</tr>
<tr>
<td>5; 322; 285; 248</td>
</tr>
<tr>
<td>6; 321; 288; 247</td>
</tr>
<tr>
<td>7; 322; 291; 245</td>
</tr>
<tr>
<td>8; 321; 292; 244</td>
</tr>
<tr>
<td>9; 320; 293; 243</td>
</tr>
<tr>
<td>10; 319; 294; 242</td>
</tr>
</tbody>
</table>

Le fichier d’entrée ayant une structure différente de celle requise pour l’utilisation d’un tLineChart, un tMap sera utilisé dans ce scénario, afin de mettre en correspondance les données avec un fichier CSV ayant la structure requise, avant d’utiliser le tLineChart pour générer un fichier représentant le graphique en lignes.

Remarque :
Le tMap est généralement utilisé afin d’adapter le schéma d’entrée, conformément à la structure du schéma du composant tLineChart. Pour plus d’informations concernant le composant tMap, consultez le Guide utilisateur du Studio Talend ainsi que le composant tMap à la page 2113.

Déposer et relier les composants

Procédure

1. A partir de la Palette, déposez les composants suivants dans l’espace de modélisation graphique :
deux composants tFileInputDelimited, un tMap, trois composants tFileOutputDelimited et un tLineChart.
2. Connectez le premier composant tFileInputDelimited au tMap à l’aide d’une connexion Row > Main.
3. Reliez le tMap au premier tFileOutputDelimited à l’aide d’une connexion Row > Main et appelez-la TV_A.
4. Répétez l’étape ci-dessus : connectez le tMap aux deux composants tFileOutputDelimited restants à l’aide d’une connexion Row > Main et appelez-les respectivement TV_B et TV_C.
6. Connectez le premier composant tFileInputDelimited au second tFileInputDelimited à l’aide d’une connexion Trigger > OnSubjobOk.
7. Connectez le premier tFileDelete au premier composant tFileInputDelimited, et le second tFileInputDelimited au second tFileDelete à l’aide d’une connexion Trigger > OnSubjobOk.
8. Renommez les composants comme vous le souhaitez afin qu’ils décrivent au mieux leur fonction.
Lire les données source

Procédure

1. Double-cliquez sur le premier composant **tFileInputDelimited** afin d’afficher l’onglet **Basic settings**.

2. Renseignez le chemin d’accès au fichier à traiter dans le champ **File name** en parcourant votre répertoire et en sélectionnant le fichier d’entrée.

3. Spécifiez dans le champ **Header** le nombre de lignes d’en-tête, une dans ce scénario. Laissez les autres paramètres tels qu’ils sont.

4. Cliquez sur **Edit schema** pour décrire la structure des données du fichier d’entrée. Dans ce scénario, le schéma d’entrée est composé de quatre colonnes : **Week**, **Mins_TVA**, **Mins_TVB**, et **Mins_TVC**. Une fois le nom des colonnes et les types de données définis, cliquez sur **OK** pour fermer la boîte de dialogue du schéma.
Adapter les données source au schéma tLineChart

Procédure

1. Double-cliquez sur le tMap pour ouvrir l'éditeur Map Editor.
   Le panneau de sortie contient une table d'entrée, ici row1, et le panneau de sortie contient trois tables vides nommées TV_A, TV_B et TV_C.

2. Utilisez l'éditeur Schema editor pour ajouter trois colonnes à chacune des tables de sortie : series (type String), x (type Integer) et y (type Integer).


4. Déposez la colonne Week de la table d'entrée dans la colonne x de chaque table de sortie.

5. Déposez la colonne Mins_TVA de la table d'entrée dans la colonne y de la table TV_A.

6. Déposez la colonne Mins_TVB de la table d'entrée dans la colonne y de la table TV_B.

7. Déposez la colonne Mins_TVC de la table d'entrée dans la colonne y de la table Mins_TV_C.
8. Cliquez sur OK pour sauvegarder le mapping, fermer l’éditeur et propager le schéma de sortie au composant de sortie.

**Générer le fichier d’entrée temporaire**

**Procédure**

1. Double-cliquez sur le premier composant `tFileOutputDelimited` pour afficher l’onglet `Basic settings`.

2. Dans le champ **File Name**, définissez un fichier CSV dans lequel les flux de données mappés seront envoyés. Dans cet exemple, nommez le fichier de sortie à créer `Temp.csv`. Ce fichier sera utilisé comme fichier d’entrée pour le composant `tLineChart`.

3. Cochez la case **Append**.

4. Définissez les paramètres des deux autres composants `tFileOutputDelimited` exactement de la même manière que dans le premier `tFileOutputDelimited`.

**Remarque :**

Notez que l’ordre des flux de données sortant du composant `tMap` n’est pas nécessairement l’ordre dans lequel les données seront écrites dans le composant cible. Pour vous assurer que le fichier est correctement généré, supprimez, s’il existe, le fichier portant le même nom.
nom avant d’exécuter le Job. Vérifiez que la case **Append** est cochée dans tous les composants **tFileOutputDelimited** afin que tous les flux de données mappés aillent dans le même fichier, sans écraser les données existantes.

**Configurer le tLineChart**

**Procédure**

1. Double-cliquez sur le second composant **tFileInputDelimited** afin d’afficher son onglet **Basic settings**.

2. Saisissez, dans le champ **File name**, le chemin d’accès au fichier, ainsi que son nom, le même que celui défini dans chaque composant **tFileOutputDelimited**. Dans cet exemple, le fichier d’entrée du **tLineChart** est **Temp.csv**.

3. Double-cliquez sur le composant **tLineChart** pour afficher l’onglet **Basic settings**.

5. Le schéma d’entrée et de sortie doivent être synchronisés. Si besoin est, copiez toutes les colonnes du schéma de sortie dans le schéma d’entrée en cliquant sur la double-flèche pointant vers la gauche. Cliquez sur OK pour fermer la boîte de dialogue du schéma.

6. Dans le champ Generated image path, définissez le chemin d’accès du fichier image à générer.

7. Dans le champ Chart title, saisissez un titre pour le graphique en lignes. Dans ce scénario, son titre est Average Weekly Viewing (per person).

8. Saisissez un nom pour les axes (X) et (Y). Dans ce scénario, le nom des axes est respectivement Week et Minutes.

9. Définissez comme vous le souhaitez la taille de l’image (dans les champs Image width pour la largeur et Image height pour la hauteur), la période dans laquelle est calculée la moyenne mobile (dans le champ Period une fois la case Moving average cochée), les limites inférieures et supérieures (respectivement dans les champs Lower bound et Upper bound), ainsi que la couleur de fond du graphique et du plan (respectivement dans les champs Chart background et Plot background).

Dans ce scénario, définissez la taille de l’image à 450 by 450, paramétrez le champ Lower bound à 210 et Upper bound à 340, sélectionnez Light_gray dans la liste Chart background et laissez les autres paramètres tels qu’ils sont.
Supprimer le fichier temporaire

Procédure

1. Double-cliquez sur le premier composant **tFileDelete** pour afficher sa vue **Basic settings**.

![Del_temp(tFileDelete_1)](image)

- **File Name**: *D:/Input/Temp.csv*
- **Fail on error**, **Delete Folder**, **Delete File Or Folder**

2. Dans le champ **File name**, renseignez le chemin d'accès du fichier d'entrée et décochez la case **Fail on error** pour permettre au Job principal d'être exécuté si le fichier à supprimer n'existe pas.
3. Spécifiez le même chemin d'accès que dans l'autre composant **tFileDelete**.

Exécuter le Job

Procédure

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** pour l'exécuter.

Résultats

Un graphique en lignes est généré selon les critères définis. Il affiche une comparaison du temps de visionnage moyen et des évolutions dans le temps correspondant aux différentes chaînes de télévision.
tLogCatcher

Ce composant fonctionne comme une fonction de log déclenchée par une des trois fonctions suivantes : Java exception, tDie ou tWarn, pour collecter et transférer des données de log.

Les composants tDie et tWarn sont étroitement liés au composant tLogCatcher. Ils sont généralement utilisés ensemble afin que les données de log collectées par le tLogCatcher soient rassemblées et envoyées vers la sortie définie.

Le composant tLogCatcher récupère les champs et les messages des fonctions Java Exception et des composants tDie et/ou tWarn et les transmet au composant suivant.

Propriétés du tLogCatcher Standard

Ces propriétés sont utilisées pour configurer le tLogCatcher s’exécutant dans le framework de Jobs Standard.

Le composant tLogCatcher Standard appartient à la famille Logs & Errors.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Le schéma de ce composant est en lecture seule. Ce schéma décrit les propriétés des données de log. Vous pouvez cliquer sur le bouton [...] à côté de Edit schema, afin de voir le schéma prédéfini contenant les champs suivants :</td>
</tr>
<tr>
<td></td>
<td>• moment : le moment où un message est capturé.</td>
</tr>
<tr>
<td></td>
<td>• pid : l’identifiant de processus du Job.</td>
</tr>
<tr>
<td></td>
<td>• root_pid : l’identifiant de processus racine.</td>
</tr>
<tr>
<td></td>
<td>• father_pid : l’identifiant de processus père.</td>
</tr>
<tr>
<td></td>
<td>• project : le nom du projet.</td>
</tr>
<tr>
<td></td>
<td>• job : le nom du Job.</td>
</tr>
<tr>
<td></td>
<td>• context : le contexte utilisé pour exécuter le Job.</td>
</tr>
<tr>
<td></td>
<td>• priority : le niveau de priorité du message.</td>
</tr>
<tr>
<td></td>
<td>• type : le type du message.</td>
</tr>
<tr>
<td></td>
<td>• origin : le nom du composant qui déclenche le message.</td>
</tr>
<tr>
<td></td>
<td>• message : le contenu du message.</td>
</tr>
<tr>
<td></td>
<td>• code : le niveau du code d’erreur.</td>
</tr>
</tbody>
</table>

| Catch Java Exception | Cochez cette case pour déclencher la fonction tCatch lorsque la fonction Java Exception est appelée dans le Job. |
**tLogCatcher**

| Catch tDie | Cochez cette case pour déclencher la fonction tCatch lorsque le composant tDie est appelé dans le Job. |
| Catch tWarn | Cochez cette case pour déclencher la fonction tCatch lorsque le composant tWarn est appelé dans le Job. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de début dans un Job secondaire qui est déclenché automatiquement à la fin du Job principal. |

**Capturer les messages déclenchés par un composant tWarn**

Cet exemple vous montre comment utiliser le composant tLogCatcher afin de capturer les messages déclenchés par un composant tWarn.

**Créer un Job afin de capturer les messages déclenchés par un composant tWarn**

Créez un Job afin de déclencher des messages à l’aide du composant tWarn, de capturer ensuite les messages en utilisant le composant tLogCatcher et d’afficher les messages dans la console.
Procédure

1. Créez un nouveau Job et ajoutez un composant tRowGenerator, un composant tWarn, un composant tLogCatcher et un composant tLogRow en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la Palette.

2. Reliez le composant tRowGenerator au composant tWarn à l'aide d'un lien Row > Main.

3. Reliez le composant tLogCatcher au composant tLogRow à l'aide d'un lien Row > Main.

Configurer un Job pour capturer les messages déclenchés par un composant tWarn

Configurez les composants dans le Job capturant les messages déclenchés par le composant tWarn et affichant par la suite les messages dans la console.

Procédure

1. Double-cliquez sur le composant tRowGenerator afin d'ouvrir l'éditeur Row Generator.

2. Définissez le schéma en ajoutant une colonne id de type Integer et sélectionnez la fonction Numeric.sequence(String,int,int) prédéfinie dans la colonne Functions.
3. Saisissez le nombre de lignes à générer dans le champ **Number of Rows for RowGenerator**, 3 dans cet exemple. Cela fait, cliquez sur **OK** pour fermer la boîte de dialogue.

4. Double-cliquez sur le composant **tWarn** afin d’ouvrir sa vue **Basic settings**.

5. Sélectionnez **Info** dans la liste déroulante **Priority**.

6. Dans le champ **Warn message**, saisissez le message à déclencher lorsqu’un nouvel enregistrement est généré, *a new record is generated* dans cet exemple.

7. Double-cliquez sur le composant **tLogCatcher** pour ouvrir sa vue **Basic settings**.

8. Cochez la case **Catch tWarn** afin de capturer les messages déclenchés par un composant **tWarn**.

9. Double-cliquez sur le composant **tLogRow** afin d’ouvrir sa vue **Basic settings**. Sélectionnez ensuite **Table (print values in cells of a table)** dans la zone **Mode** pour obtenir une meilleure lisibilité des résultats.

**Exécuter un Job pour capturer les messages déclenchés par un composant **tWarn****

Une fois le Job et ses composants configurés pour capturer les messages déclenchés par le composant **tWarn**, vous pouvez exécuter le Job et vérifier les résultats de son exécution.

**Procédure**

1. Appuyez sur **Ctrl + S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour l’exécuter.
Comme affiché ci-dessus, tous les messages déclenchés sont collectés et affichés dans la console.

**Capturer le message déclenché par un composant tDie**

Cet exemple vous montre comment utiliser un composant **tLogCatcher** pour capturer le message déclenché par un composant **tDie**.

**Créer un Job pour capturer le message déclenché par un composant tDie**

Créez un Job afin de déclencher un message à l’aide du composant **tDie**, de capturer ensuite le message en utilisant le composant **tLogCatcher** et d’afficher le message dans la console.

**Procédure**

1. Créez un nouveau Job et ajoutez un composant **tRowGenerator**, un **tFileOutputDelimited**, un **tDie**, un **tJava**, un **tLogCatcher** et un composant **tLogRow** en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le composant **tRowGenerator** au composant **tFileOutputDelimited** à l’aide d’un lien **Row > Main**.
3. Reliez le composant **tFileOutputDelimited** au **tDie** à l’aide d’un lien **Trigger > Run if**.
4. Reliez le composant **tRowGenerator** au **tJava** à l’aide d’un lien **Trigger > On Subjob Ok**.
5. Reliez le composant **tLogCatcher** au composant **tLogRow** à l’aide d’un lien **Row > Main**.
Configurer un Job pour capturer le message déclenché par un composant tDie

Configurez les composants dans le Job capturant le message déclenché par le composant tDie et affichant par la suite le message dans la console.

Procédure

1. Double-cliquez sur le composant tRowGenerator afin d’ouvrir l’éditeur Row Generator.

![Row Generator](image)

2. Définissez le schéma en ajoutant une colonne id de type integer et sélectionnez la fonction Numeric.sequence(String,int,int) prédéfinie dans la colonne Functions.

3. Saisissez le nombre de lignes à générer dans le champ Number of Rows for RowGenerator, 0 dans cet exemple. Cela fait, cliquez sur OK pour fermer la boîte de dialogue.

4. Double-cliquez sur le composant tFileOutputDelimited afin d’ouvrir sa vue Basic settings. Spécifiez ensuite dans le champ File Name, le chemin d’accès au fichier qui contiendra les données à générer.

5. Cliquez sur la connexion If et, dans le champ Condition dans la vue Basic settings, spécifiez la condition déclenchant le composant tDie. Dans cet exemple, spécifiez la condition 

\[(\text{Integer})\text{globalMap.get("tRowGenerator_1_NB_LINE")}) <= 0\]

ce qui signifie que le composant tDie se déclenche lorsque le nombre de lignes à générer est inférieur ou égal à zéro.

6. Double-cliquez sur le composant tDie afin d’ouvrir sa vue Basic settings et, dans le champ Die message, saisissez le message à déclencher avant que le Job ne soit arrêté. Dans cet exemple, no row generated.
7. Double-cliquez sur le composant **tJava** afin d’ouvrir sa vue **Basic settings** et, dans le champ **Code**, saisissez

```
System.out.println("The number of rows generated is "+((Integer)globalMap.get("tRowGenerator_1_NB_LINE"))+". #This message will not be displayed if no row is generated.");
```

Notez que ce message s’affiche uniquement lorsque le nombre de lignes généré est supérieur à zéro. Dans cet exemple, le nombre de lignes à générer est 0 : le Job sera donc arrêté et ce message ne s’affichera pas.

8. Double-cliquez sur le composant **tLogCatcher** afin d’ouvrir sa vue **Basic settings** et cochez la case **Catch tDie** pour capturer le message déclenché par le **tDie**.

9. Double-cliquez sur le composant **tLogRow** afin d’ouvrir sa vue **Basic settings**, sélectionnez **Table (print values in cells of a table)** dans la zone **Mode** pour obtenir une meilleure lisibilité du résultat.

**Exécuter un Job pour capturer le message déclenché par un composant tDie**

Une fois le Job et ses composants configurés pour capturer le message déclenché par le composant **tDie**, vous pouvez exécuter le Job et vérifier les résultats de son exécution.

**Procédure**

1. Appuyez sur **Ctrl + S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour l’exécuter.

Comme affiché ci-dessus, le message déclenché est collecté et affiché dans la console.
3. Modifiez le nombre de lignes à générer (10 par exemple) dans le champ **Number of Rows for RowGenerator** du composant **tRowGenerator**, sauvegardez et exécutez le Job une nouvelle fois pour obtenir un résultat différent.
tLogRow

Ce composant affiche les données ou les résultats dans la console Run afin de montrer les données traitées.

Propriétés du tLogRow Standard

Ces propriétés sont utilisées pour configurer le tLogRow s’exécutant dans le framework de Jobs Standard.

Le composant tLogRow Standard appartient à la famille Logs & Errors.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
<tr>
<td>Sync columns</td>
<td>Cliquez pour synchroniser le fichier de sortie avec le schéma du fichier d’entrée. Cette fonction n’est disponible que quand le composant est relié au composant précédent via un lien Row.</td>
</tr>
<tr>
<td>Basic</td>
<td>Affiche le flux de sortie en mode basique.</td>
</tr>
<tr>
<td>Table</td>
<td>Affiche le flux de sortie dans des cellules de table.</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>Vertical</td>
<td>Affiche chaque ligne du flux de sortie comme une liste de valeurs clé. Quand cette option est activée, vous pouvez, pour chaque ligne de sortie, choisir entre le nom unique, le nom du composant et les deux.</td>
</tr>
<tr>
<td>Separator</td>
<td>Saisissez le séparateur qui délimitera les données dans la console.</td>
</tr>
<tr>
<td>Print header</td>
<td>Activez cette option pour inclure l’en-tête du flux d’entrée dans l’affichage de sortie.</td>
</tr>
<tr>
<td>Print component unique name in front of each output row</td>
<td>Cochez cette case si plusieurs composants <code>tLogRow</code> sont utilisés. Cela permet de différencier les sorties.</td>
</tr>
<tr>
<td>Print schema column name in front of each value</td>
<td>Cochez cette case pour récupérer les libellés des colonnes du schéma de sortie.</td>
</tr>
<tr>
<td>Use fixed length for values</td>
<td>Cochez cette case pour paramétrer une largeur fixe aux valeurs affichées.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

| Global Variables | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable `After` et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case `Die on error` est décochée, si le composant a cette option. Une variable `Flow` fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |
|-------------------|---------------------------------------------------------------|

2106
## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en composant intermédiaire dans un flux de données ou en composant de fin dans un Job design.</th>
</tr>
</thead>
</table>

### Scénarios associés

Pour plus d’informations sur l’utilisation du **tLogRow**, consultez les composants :

- **tFileInputDelimited**, [Lire les données maître depuis un hub MDM](#) à la page 2276.
- **tContextLoad**, [Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement](#) à la page 520.
- **tWarn**, **tDie** et **tLogCatcher**, [Capturer les messages déclenchés par un composant tWarn](#) à la page 2098 et [Capturer le message déclenché par un composant tDie](#) à la page 2101.
tLoop

Ce composant exécute automatiquement une tâche ou un Job dans une boucle.

Propriétés du tLoop Standard

Ces propriétés sont utilisées pour configurer le tLoop s'exécutant dans le framework de Jobs Standard.

Le composant tLoop Standard appartient à la famille Orchestration.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Loop Type</th>
<th>Sélectionnez le type de boucle à effectuer : For ou While.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>For : La tâche ou le Job est exécuté(e) pour le nombre de boucle défini.</td>
</tr>
<tr>
<td></td>
<td>While : La tâche ou le Job est exécuté(e) jusqu’à ce que la condition soit rencontrée.</td>
</tr>
</tbody>
</table>

For

- From : saisissez le numéro de la première instance sur laquelle la boucle doit commencer. Si vous sélectionnez l’instance numéro 2 comme instance de départ et que vous prenez un écart de 2, cela signifie que la boucle se fera sur tous les nombres pairs.
- To : saisissez la dernière instance sur laquelle la boucle doit passer.
- Step : saisissez l’écart entre deux instances avec lequel la boucle doit s’effectuer. Un écart de 2 signifie que la boucle passe toutes les deux instances.
- Values are increasing : cochez cette case pour autoriser uniquement une séquence croissante.

While

- Declaration : saisissez une expression déclenchant la boucle.
- Condition : saisissez la condition devant être rencontrée pour que la boucle s’arrête.
- Iteration : saisissez l’expression décrivant l’opération à effectuer à chaque boucle.

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
Global Variables

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

**CURRENT_VALUE** : valeur courante. Disponible uniquement pour les boucles de type *For*. Cette variable est une variable *Flow* et retourne un entier.

**CURRENT_ITERATION** : numéro de séquence de l’itération courante. Cette variable est une variable *Flow* et retourne un entier.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

Utilisation

**Règle d’utilisation**

Le composant *tLoop* est un composant de début et requiert une connexion de type *Iterate* au composant suivant.

<table>
<thead>
<tr>
<th>For</th>
<th>From</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>To</td>
</tr>
<tr>
<td>Step</td>
<td></td>
</tr>
</tbody>
</table>

**Values are increasing**

**While**

**Declaration**

**Condition**

**Iteration**

**Connections**

Liens de sortie (de ce composant à un autre) :

*Row* : Iterate.


Liens d’entrée (d’un autre composant à celui-ci) :

*Row* : Iterate.
**Scénario : Exécuter un Job à partir d’une boucle**

Ce scénario décrit un Job constitué d’un Job parent et d’un Job enfant. Le Job parent exécute une boucle qui permet d’exécuter un Job enfant cinq fois, avec une pause de trois secondes entre chaque exécution.

**Procédure**

1. Créez un Job nommé *Parent* puis déposez un *tLoop*, un *tRunJob* et un *tSleep* dans l’espace de modélisation graphique.
2. Reliez le *tLoop* au *tRunJob* l’aide d’un lien Row > Iterate puis reliez le *tRunJob* au *tSleep* à l’aide d’un lien Row > Main.

3. Créez un Job nommé *Child* puis déposez un *tRowGenerator* et un *tLogRow* dans l’espace de modélisation graphique.
4. Reliez le *tRowGenerator* au *tLogRow* à l’aide d’un lien Row > Main.

5. Dans la vue Basic settings du composant *tLoop*, définissez le type de boucle (For) puis saisissez le numéro de l’instance sur lequel commencer (1), le numéro de l’instance sur lequel finir (5) et l’écart à utiliser (1).

---


Pour plus d’informations concernant les liens, consultez le Guide utilisateur du Studio Talend.
6. Dans la vue **Basic settings** du composant **tRunJob**, sélectionnez le Job *Child* dans le champ **Job**.

7. Dans la vue **Basic settings** du composant **tSleep**, saisissez la durée de la pause en secondes dans le champ **Pause**. Pour cet exemple, la pause est de 3 secondes.

8. Double-cliquez sur le composant **tRowGenerator** afin d'ouvrir l'éditeur de schéma. Cliquez ensuite sur le bouton **[+]** afin d'ajouter quatre colonnes :
   - *id*, afin de générer les numéros de séquence
   - *firstname*, afin de générer des prénoms aléatoires
   - *lastname*, afin de générer des noms aléatoires
   - *city*, afin de générer des noms de ville aléatoires

Pour plus d’informations concernant le **tRowGenerator**, consultez **tRowGenerator** à la page 3478.
9. Appuyez sur F6 afin d'exécuter le Job *Parent*.

Le Job *Child* est exécuté cinq fois avec une pause de trois secondes entre chacune des deux exécutions. Comme configuré dans le composant `tRowGenerator`, des informations personnelles aléatoires sont affichées dans la console *Run*. 
tMap

Ce composant transforme et dirige les données à partir d’une ou plusieurs source(s) et vers une ou plusieurs destination(s).

Le tMap est un composant avancé qui s’intègre au Studio Talend comme un plug-in.

Conseil :
Il n’y a pas d’ordre dans les flux de sortie du tMap. Pour exécuter les flux de sortie un par un, vous pouvez les écrire en sortie dans des fichiers temporaire ou dans la mémoire, puis les lire et les insérer dans des fichiers ou bases de données à l’aide de différents sous-jobs reliés par des liens Trigger > OnSubjobOK.

Propriétés du tMap Standard

Ces propriétés sont utilisées pour configurer le tMap s’exécutant dans le framework de Jobs Standard.
Le composant tMap Standard appartient à la famille Processing.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Map editor</th>
</tr>
</thead>
</table>

Le Mapper est l’éditeur du tMap. Il vous permet de définir les propriétés d’aiguillage et de transformation des données.

Si nécessaire, cliquez sur le bouton [Property Settings], qui propose les options suivantes :

• Die on error : Cochez cette case si vous souhaitez arrêter le Job en cas d’erreur. Cette case est cochée par défaut.

• Enable Auto-Conversion of types : Si vos colonnes d’entrée et de sortie dans un mapping ne sont pas du même type de données, cochez cette case afin de permettre une conversion automatique du type lors de l’exécution et d’éviter les erreurs de compilation.

Cette option est activée par défaut si la case Enable Auto-Conversion of types est cochée dans la vue Project Settings lorsque le composant est ajouté. Vous pouvez également écraser le comportement de conversion par défaut de ce composant en configurant des règles de conversion dans la vue Project Settings. Pour plus d’informations, consultez le Guide utilisateur du Studio Talend.

Notez que la conversion automatique entre les types Date et BigDecimal n’est pas supportée.

• Store on disk : Les options disponibles dans cette zone sont identiques aux options pertinentes disponibles respectivement dans les onglets Basic settings et Advanced settings. Les paramètres
choisis dans la boîte de dialogue [Property Settings] sont reflétés dans les vues de l’onglet respectif et vice versa.

### Mapping links display as
- **Auto** : par défaut, les liens sont en forme de courbes.
- **Curves** : les liens du mapping sont en forme de courbes.
- **Lines** : les liens du mapping sont en forme de lignes droites. Cette dernière option améliore légèrement les performances.

### Temp data directory path
Saisissez le chemin du dossier dans lequel vous souhaitez stocker les données temporaires générées par le traitement des données de référence (lookup). Pour plus d’informations sur ce dossier, consultez le Guide d’utilisation du Studio Talend.

### Preview
L’aperçu est un instantané des données du Mapper. Il n’est visible que lorsque les propriétés du Mapper sont renseignées. La synchronisation de l’aperçu ne prend effet qu’après la sauvegarde des modifications.

### Advanced settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max buffer size (nb of rows)</td>
<td>Saisissez la taille de la mémoire physique, en nombre de lignes, que vous souhaitez allouer aux données traitées.</td>
</tr>
<tr>
<td>Ignore trailing zeros for BigDecimal</td>
<td>Cochez cette case pour ignorer les zéros non nécessaires dans le traitement de données BigDecimal.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Plusieurs utilisations sont possibles, de la simple réorganisation des champs de données aux transformations les plus complexes, telles que le multiplexage et le démultiplexage de données, la concaténation, l’inversion, le filtrage, etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>L’utilisation du composant <strong>tMap</strong> requiert un niveau moyen de connaissances en Java afin d’exploiter au mieux ses fonctionnalités. Ce composant est utilisé comme une étape intermédiaire, c’est pourquoi il ne peut pas être utilisé comme composant de début ou de fin de Job.</td>
</tr>
</tbody>
</table>

Mapping simple de données avec filtre et jointure explicite

Le Job ci-dessous a pour objectif de lire des données d’un fichier .csv stocké dans le **Repository**, de rechercher des données d’un fichier de référence, dont le schéma est également stocké dans le **Repository** puis d’extraitre des données de ces deux fichiers en fonction de filtres et d’envoyer ces données vers les fichiers de sortie et de rejet.

**Construire le Job**

**Procédure**

1. Déposez deux composants **tFileInputDelimited**, un **tMap** et trois **tFileOutputDelimited** de la Palette dans l’espace de modélisation graphique.
2. Renommez respectivement les deux **tFileInputDelimited Cars** (voitures) et **Owners** (propriétaires).
3. Reliez les deux composants d’entrée au **tMap** à l’aide de liens **Row > Main** et nommez ces connexions **Cars_data** et **Owners_Data**, respectivement.
4. Reliez le **tMap** aux trois composants de sortie à l’aide de liens **Row > New Output (Main)** et nommez ces connexions, respectivement, **Insured**, **Reject_NoInsur** et **Reject_OwnerID**.
Configurer les composants

Procédure

1. Double-cliquez sur le composant `tFileInputDelimited` nommé `Cars` pour paramétrer ses propriétés de base (Basic settings).

2. Sélectionnez Repository dans les champs Property type et Schema. Pour le schéma, sélectionnez la métdonnée `cars` dans la boîte de dialogue [Repository Content]. Le reste des champs est automatiquement renseigné lorsque vous sélectionnez la métdonnée appropriée dans la liste.

3. Double-cliquez sur le composant `Owners` et définissez à nouveau les propriétés. Sélectionnez la métdonnée appropriée, `owners` dans cet exemple.

   Remarque :
   
   Dans ce scénario, les schémas d’entrée sont stockés dans le nœud Metadata de la vue Repository afin de les retrouver plus facilement. Pour plus d’informations concernant la création de métdonnées dans le Repository, consultez le Guide utilisateur du Studio Talend.

4. Double-cliquez sur le composant `tMap` pour ouvrir le Mapper. La zone Input (données en entrée) est déjà renseignée avec les schémas des composants d’entrée et la première table correspond au
flux principal (Main Row). Les libellés de chacune des connexions d’entrée apparaissent en en-tête des tables.

5. Créez une jointure entre les deux tables d’entrée en faisant simplement glisser le champ ID_Owner de la table principale (Cars_data) vers le champ équivalent de la table de référence (Owners_data).

6. Définissez le lien en Inner Join en cliquant sur le bouton tMap settings, en cliquant dans le champ Value pour Join Model, en cliquant sur le bouton qui apparaît dans le champ et en sélectionnant Inner Join dans la liste Options.

7. Glissez et déposez le contenu de la table Cars_data vers la table Insured.

8. Glissez les colonnes ID_Owner, Registration et ID_Reseller de la table Cars_data, ainsi que la colonne Name de la table Owners_data dans la table Reject_NoInsur.

9. Glissez toutes les colonnes de la table Cars_data dans la table Reject_OwnerID.
   Pour plus d’informations concernant le mapping de données, consultez le Guide utilisateur du Studio Talend.

10. Cliquez sur le bouton [+ ] en haut de la table Insured afin d’ajouter une ligne de filtre.
    Glissez la colonne ID_Insurance de la table Owners_data vers la zone de filtre et saisissez la formule suivante d’exclusion des valeurs non définies : Owners_data.ID_Insurance != null.
    Avec ce filtre, la table Insured va rassembler tous les enregistrements qui possèdent un ID Insurance.
11. Cliquez sur le bouton **tMap settings** en haut de la table **Reject_NoInsur** et sélectionnez **true** pour **Catch lookup inner join reject** afin que de définir cette table comme le flux de rejet standard pour rassembler les enregistrements n’ayant pas d’ID Insurance.

12. Cliquez sur le bouton **tMap Settings** en haut de la table **Reject_OwnerID** puis sélectionnez **true** pour **Catch lookup inner join reject** afin que cette table rassemble les enregistrements du flux Cars_data ayant des IDs manquants ou ne correspondant pas.
Cliquez sur **OK** pour valider et revenir à l'espace de modélisation.

**13.** Double-cliquez sur chacun des composants de sortie afin de définir leurs propriétés. Si vous souhaitez créer un nouveau fichier, parcourez votre système jusqu’au dossier de destination et saisissez le nom du fichier avec son extension.

Cochez la case **Include header** pour réutiliser les libellés des colonnes du schéma d’entrée dans comme en-tête dans votre fichier de sortie.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** pour exécuter le Job.

Les fichiers de sortie sont créés et ils contiennent les données définies.

**Mapping de données avec rejet Inner join**

Ce scénario, basé sur le scénario 1, requiert l’ajout d’un fichier d’entrée contenant les informations sur les revendeurs, ainsi que l’ajout de champs supplémentaires dans la table de sortie principale. De plus, deux filtres sont ajoutés sur les jointures Inner Join afin de récupérer les rejets spécifiques.
**Construire le Job**

**Procédure**

1. Déposez un **tFileInputDelimited** et un **tFileOutputDelimited** dans l'espace de modélisation graphique et renommez les composants, respectivement *Resellers* et *No_Reseller_ID*.
2. Connectez le **tFileInputDelimited** au **tMap** à l'aide d'un lien **Row > Main** et renommez le lien *Resellers_data*.
3. Connectez le **tMap** au nouveau **tFileOutputDelimited** à l'aide d'un lien **Row > New Output**. Renommez le lien *Reject_ResellerID*.

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le composant *Resellers* pour définir ses propriétés dans la vue **Basic settings**.
2. Sélectionnez Repository dans la liste Property Type et sélectionnez le schéma resellers, dans la boîte de dialogue [Repository Content]. Les autres champs sont automatiquement renseignés.

   **Remarque :**
   
   Dans ce scénario, les schémas d'entrée sont stockés sous le nœud Metadata du Repository afin de les retrouver facilement. Pour plus d'informations concernant la création de métadonnées dans le Repository, consultez le Guide utilisateur du Studio Talend.

3. Double-cliquez sur le composant tMap pour ouvrir le Map Editor. Vous remarquerez que le schéma est automatiquement ajouté à la zone Input des schémas d’entrée.

4. Créez une jointure entre le flux d’entrée principal et le nouveau flux d’entrée en déposant la colonne ID_Reseller de la table Cars_data dans la colonne ID_Reseller de la table Resellers_data.

5. Cliquez sur le bouton tMap settings en haut de la table Resellers_data et sélectionnez Inner Join comme modèle de jointure (Join Model).

Remarque :
Lorsque deux Inner Joins sont définis, vous devez soit définir deux tables de rejet inner join pour différencier les deux rejets, soit rassembler les deux rejets inner join dans une seule et même table de sortie, s'il n'y a qu'une seule sortie.

7. Dans la zone de sortie Output, cliquez sur le signe [+] pour ajouter une nouvelle table de sortie et nommez-la Reject_ResellerID.

8. Déposez toutes les colonnes de la table Cars_data dans la table Reject_ResellerID.

9. Cliquez sur le bouton tMap Settings puis sélectionnez true pour Catch lookup inner join reject afin de définir la dernière table comme étant la table du flux de sortie de rejets Inner Join. Si la jointure Inner Join ne peut être établie pour une entrée, les informations concernant les voitures correspondantes seront rassemblées dans le flux de sortie.
10. Appliquez maintenant un filtre sur chacune des sorties de rejet Inner Join afin de distinguer les deux types de rejets.
   Dans la première table de rejet (Reject.OwnerID), cliquez sur la flèche dotée d’un signe [+]] pour ajouter une ligne de filtre et renseignez ce champ avec la formule suivante pour ne recueillir que les rejets liés au champ OwnerID non défini : Owners.data.ID_Owner==null

11. Dans la seconde table de rejet Inner Join (Reject.ResellerID), renouvelez l’opération et utilisez la formule suivante : Resellers.data.ID_Reseller==null

Cliquez sur OK pour enregistrer les changements et fermer le Mapper.

12. Double-cliquez sur le composant No Reseller ID pour afficher sa vue Basic settings.

Dans ce scénario, spécifiez simplement le chemin d’accès au fichier de sortie et cochez la case Include Header. Laissez les autres paramètres tels qu’ils sont.

13. Pour comprendre le fonctionnement du Mapper, dans ce scénario, retirez du fichier Resellers.csv les lignes correspondant aux Reseller ID 5 et 8.
Exécuter le Job

Procédure

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** pour l’exécuter.

Résultats

Les quatre fichiers de sortie sont tous créés dans le dossier défini dans les propriétés de la vue **Component**. Le fichier **No_Reseller_ID.csv**, *NoResellerID.csv*, le champ **ID_Owners** correspond aux entrées du fichier **Cars** dont le Reseller ID est égal à 5 et 8, manquant dans le fichier **Resellers.csv**.

Un troisième scénario plus avancé et basé sur le scénario 2 requiert l’ajout d’une nouvelle entrée contenant les informations d’assurance, par exemple.

Dans le Mapper, mettez en place une jointure **Inner Join** entre deux tables de référence (Lookup) des flux secondaires **Owners** et **Insurance** afin de gérer les références en cascade et par conséquent de récupérer les informations de la table **Insurance** via les données de la table **Owners**.

Mapping avancé avec filtres, jointures explicites et rejet **Inner join**

Ce scénario présente un Job utilisant le **tMap** et permettant de trouver les clients propriétaires d’une voiture de marque définie et avoir entre 2 et 6 enfants (inclus) dans un but de vente incitative.

Construire le Job

Procédure

1. Cliquez-déposez les composants suivants de la Palette dans l’espace de modélisation graphique : trois **tFileInputDelimited**, un **tMap** et deux **tFileOutputDelimited**.
2. Connectez les composants d’entrée au **tMap** à l’aide de connexions de type **Main row**.
   Faites attention au premier composant que vous connectez car son flux de données sera automatiquement un flux **Main** (principal). Et toutes les autres connexions seront alors de type **Lookup** (flux de référence). Dans ce scénario, le composant d’entrée **Owners** sera relié au **tMap** par le flux **Main**.
Configurer les composants

Procédure

1. Définissez les propriétés de chaque composant d’entrée dans l’onglet **Basic settings**. Configurez les propriétés du composant **Owners**.

2. Sélectionnez **Repository** dans la liste **Property Type** et sélectionnez le schéma **resellers**, dans la boîte de dialogue [**Repository Content**]. Les autres champs sont automatiquement renseignés.

   **Remarque** :
   
   Dans ce scénario, les schémas d’entrée sont stockés sous le nœud **Metadata** du **Repository** afin de les retrouver facilement. Pour plus d’informations concernant la création de métadonnées dans le **Repository**, consultez le *Guide utilisateur du Studio Talend*.

   Répétez ces étapes pour les autres composants d’entrée : **Cars** et **Resellers**. Ces deux flux de référence (**Lookup**) renseigneront les tables secondaires (lookup) de la zone d’entrée du **Map Editor**.

3. Double-cliquez sur le composant **tMap** pour ouvrir le **Map Editor** et définir le mapping et les filtres.
Configurez une jointure explicite entre le flux Main Owner et le flux Lookup Cars en glissant-déposant la colonne ID_Owner de la table Owner dans la colonne ID_Owner de la table Cars. La jointure explicite s'affiche avec une clé de hachage.

4. Dans le champ Expr. Key de la colonne Make (Marque), saisissez (en Java) le filtre à appliquer. Dans ce cas d'utilisation, saisissez simplement "BMW" puisque ce qui est recherché ce sont les propriétaires (Owners) de cette marque en particulier.

5. Effectuez une jointure en cascade entre les deux tables de référence Cars et Resellers, sur la colonne ID_Reseller, afin de retrouver les informations sur les propriétaires et plus précisément le nombre d’enfants qu’ils ont.

6. Comme vous souhaitez exclure les valeurs nulles dans une table séparée et les exclure de la sortie standard, cliquez sur le bouton tMap Settings puis sélectionnez Inner Join comme modèle de jointure pour chaque table Lookup à filtrer.
7. Dans les paramètres du tMap, vous pouvez sélectionner une des options suivantes : **Unique match**, **First match**, ou **All Matches**. Dans cet exemple, sélectionnez l’option **All matches**. Ainsi, si plusieurs correspondances sont trouvées dans l’Inner Join (i.e. les lignes répondant à la jointure explicite ainsi qu’au filtre), elles seront toutes prises en compte dans le flux de sortie (soit dans le flux de rejet, soit dans le flux de sortie standard).

**Remarque :**
L’option **Unique match** fonctionne de la même manière que l’option **Last match**. Les options **First match** et **All matches** fonctionnent comme l’indiquent leurs noms (respectivement première correspondance et toutes les correspondances).

8. Puis dans la zone **Output** du Map Editor, ajoutez deux tables : une pour les correspondances répondant à la jointure explicite et au filtre et une pour les rejets.

9. Déposez toutes les colonnes de la table Owners, les colonnes Registration, Make et Color de la table Cars ainsi que les colonnes ID_Reseller et Name_Reseller de la table Reseller dans la table de sortie principale.

10. Déposez toutes les colonnes de la table Owners dans la table de rejets.

11. Cliquez sur le bouton **Filter** en haut de la table de sortie principale afin d’afficher la zone d’expression **Filter**.


12. Dans la table de rejets, cliquez sur le bouton **tMap settings** et définissez les types de rejets.
 Sélectionnez true pour Catch output reject afin de collecter des données concernant les détenteurs de voitures BMW ayant moins de deux enfants ou plus de six enfants.

Sélectionnez true pour Catch lookup inner join reject afin de collecter des données concernant les propriétaires d’autres marques et ceux pour qui l’information Reseller est introuvable.

Cliquez sur OK pour valider le mapping et fermer l’éditeur.

Dans l’éditeur graphique, cliquez-droit sur le tMap et créez une connexion de sortie vers les composants correspondants.


Dans ce scénario, spécifiez simplement les chemins d’accès aux fichiers de sortie et cochez la case Include Header. Laissez les autres paramètres tels qu’ils sont.
Exécuter le Job

Procédure

1. Appuyez sur les touches Ctrl+S afin de sauvegarder votre Job.
2. Appuyez sur F6 pour l'exécuter.

Le fichier de sortie principal contient les informations concernant les détenteurs de BMW ayant entre deux et six enfants, et le fichier de rejet contient les informations concernant les autres possesseurs de voiture(s).

Mapping avancé avec filtres et différents rejets

Ce scénario est une version modifiée du scénario précédent. Il décrit un Job qui applique des filtres pour limiter la recherche aux possesseurs de voitures BMW ou Mercedes ayant entre deux et six enfants et divise les données sans correspondance en différents flux de rejets.

Construire le Job

Procédure

1. Reprenez le Job de Mapping avancé avec filtres, jointures explicites et rejet Inner join à la page 2124.
2. Déposez un nouveau tFileOutputDelimited de la Palette dans l'espace de modélisation graphique et nommez-le Rejects_BMW_Mercedes.
3. Connectez le composant tMap au nouveau composant de sortie à l'aide d'une connexion Row et nommez cette connexion selon la fonction du composant de sortie.
   Ce lien deviendra le nom de la nouvelle table de sortie dans le Map Editor.
4. Renommez les connexions et les composants de sortie existants afin de présenter leur fonction.
   Les tables de sortie dans le Map Editor seront automatiquement renommées selon le libellé des liens. Dans cet exemple, renommez les connexions de sortie BMW_Mercedes_withChildren et Owners_Other_Makes, respectivement.
Configurer les composants

Procédure

1. Double-cliquez sur le composant tMap afin de lancer le Map Editor pour modifier le mapping et les filtres.
   Remarquez que la zone de sortie contient une nouvelle table de sortie, vide, nommée Rejects BMW_Mercedes. Vous pouvez ajuster la position de la table en la sélectionnant et en cliquant sur les flèches pointant vers le haut ou vers le bas, en haut de la zone de sortie.
2. Supprimez le filtre BMW de la table Cars dans la zone d’entrée.
3. Cliquez sur le bouton Filters pour afficher le champ Filter et saisissez un nouveau filtre afin de limiter la recherche aux seules marques de voitures BMW et Mercedes. L'instruction est la suivante: `Cars.Make.equals("BMW") || Cars.Make.equals("Mercedes")`. 

4. Sélectionnez toutes les colonnes de la table principale de sortie et déposez-les dans la nouvelle table de sortie.
   Sinon, vous pouvez glisser les colonnes des tables d’entrée correspondantes dans la nouvelle table de sortie.

5. Cliquez sur le bouton tMap settings en haut de la nouvelle table de sortie et sélectionnez true pour Catch output reject afin de collecter les données concernant les détenteurs de voiture(s) BMW ou Mercedes ayant moins de deux enfants ou plus de six.

6. Dans la table Owners_Other_Makes, sélectionnez true pour Catch lookup inner join reject afin de collecter les données concernant les possesseurs de voitures d’autres marques et ceux dont l’information reseller est introuvable.
7. Cliquez sur OK pour valider le mapping et fermer le Map Editor.

8. Configurez les propriétés des composants de sortie dans leur vue Basic settings respective. Dans ce scénario, définissez simplement les nouveaux chemins d'accès aux fichiers de sortie et cochez la case Include Header.
Exécuter le Job

Procédure
1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** pour l'exécuter.

Le contenu du flux de sortie principal montre que les lignes filtrées sont bien passées.

| ID_Owner;Name;ID_Insurance;Children Nr;Registration;Make;Color;ID_Reseller;Name_Reseller |
|-----------------------------------------|-------------------------------------|
| 1: John CARTER;120;3;GQT 955;Mercedes;grey;67;Warren GARFIELD |
| 2: Benjamin ROOSEVELT;103;2;PJB 72;BMW;yellow;80;Lyndon TAFT |
| 3: Bill GRANT;70;2;TJS 558;BMW;grey;94;Grover ROOSEVELT |
| 4: Chester VAN BUREN;184;2;PZI 361;BMW;yellow;5;Benjamin HARDING |
| 5: Herbert EISENHOWER;160;3;PZS 783;Mercedes;grey;10;Theodore HARRISON |
| 6: Andrew ADAMS;16;3;GZT 196;BMW;yellow;43;William WASHINGTON |
| 7: Bill ARTHUR;145;6;REZ 45;BMW;white;43;William WASHINGTON |
| 8: Richard ROOSEVELT;3;5;GTY 395;Mercedes;yellow;55;Grover GRANT |
| 9: Ronald JEFFERSON;70;3;ZTI 771;BMW;green;51;Abraham JACKSON |
| 10: Grover KENNEDY;190;6;CITY 918;Mercedes;red;5;Harry HARRISON |

Mapping avancé avec rechargement de la base de référence à chaque ligne

Le scénario suivant décrit un Job permettant de récupérer certaines informations sur des personnes dans une base de données de référence, en se basant sur une jointure effectuée sur leur âge. Le flux principal des données source est lu à partir d’une table MySQL appelée *people_age* contenant certains détails sur les personnes, comme leur identifiant (numérique), leur nom et prénom (alphanumérique) ainsi que leur âge (numérique), qui peut être soit 40, soit 60 ans. On a intentionnellement réduit le nombre d’entrées de cette table.

Les informations de référence sont également stockées dans une table MySQL appelée *large_data_volume*. Cette table de référence contient un certain nombre d’entrées, dont le nom des villes dans lesquelles les personnes enregistrées dans la table principale sont allées. Le nombre d’entrées a été restreint pour plus de clarté mais, dans le cas d’une utilisation normale, le Job ci-dessous a des résultats plus significatifs quand on traite un gros volume de données de référence.
Pour optimiser les performances, un composant de connexion aux bases de données est utilisé au début de ce Job pour ouvrir une connexion à la base de données de référence afin de ne pas ouvrir et refermer une connexion à chaque ligne de la table de référence chargée.

Un filtre d’expression (expression filter) est appliqué à ce flux de référence, qui permet de ne sélectionner que les données des personnes dont l’âge est 60 ou 40 ans. Ainsi, seules les lignes pertinentes de la table de référence sont enregistrées pour chaque ligne du flux principal.

Ce Job présente donc comment, à partir d’un nombre limité de lignes de flux principal, on peut optimiser la base de données de référence pour n’enregistrer que les résultats correspondant à la clé d’expression (expression key).

Remarque :

En général, comme le chargement des références s’effectue pour chacune des lignes de flux principal, il est plus intéressant d’utiliser cette option quand on ne traite qu’un nombre limité de lignes de flux principal et un grand nombre de lignes de référence.

La jointure s’effectue au niveau du champ age. Les informations de la base de référence sont ensuite chargées pour chacune des lignes du flux principal en entrée, en utilisant l’option de chargement appropriée dans l’éditeur du composant tMap.

Dans cet exemple, les métadonnées ont été préparées pour le composant de connexion et les composants source. Pour plus d’informations sur les métadonnées de connexion aux bases de données et les métadonnées de création de schémas des tables, consultez la section dédiée au sujet dans le Guide utilisateur du Studio Talend.

Le Job est formé de cinq composants, quatre composants de bases de données et un composant de mapping.

Construire le Job

Procédure

1. A partir du nœud Metadata > Db Connections dans le Repository, glissez le schéma de base de données dans l’éditeur graphique. Dans cet exemple, la table source s’appelle people_age.
2. Lorsque vous déposez le composant, une liste apparaît à l’écran : sélectionnez le composant tMysqlInput.
3. Déposez le schéma de base de donnée de référence à partir du nœud Metadata > Db Connections. Dans cet exemple, la table de référence s'appelle large_data_volume.
4. De la même manière, glissez une connexion à la base de données du nœud Db Connections dans le Repository dans l'éditeur graphique. Dans la boîte de dialogue [Components], sélectionnez le composant tMysqlConnection. Ce composant crée une connexion permanente à la base de données de référence afin d'éviter que la connexion ne se referme au chargement que chaque ligne de référence.
5. A partir de la Palette, déposez dans l'éditeur graphique le composant tMap (famille Processing) et les composants tMysqlCommit et tMysqlOutput (famille Database).
6. Reliez maintenant tous les composants entre eux : faites un clic-droit sur le tMysqlInput correspondant à la table people et maintenez le bouton enfoncé tout en déplaçant le curseur jusqu'au tMap.
7. Relâchez le bouton lorsque vous êtes sur le tMap, le flux principal de lignes est créé automatiquement.
8. Renommez le lien de type Main en people, pour pouvoir identifier plus facilement le flux de données principal.
9. Effectuez la même opération pour connecter la table de référence (large_data_volume) au tMap et le tMap au tMysqlOutput.
10. Une boîte de dialogue apparaît et vous devez donner un nom au lien de sortie. Dans cet exemple, le flux de sortie s'appelle people_mixandmatch.
11. Renommez également le lien de connexion à la base de référence en large_volume pour pouvoir identifier plus facilement le flux de données.
12. Connectez le tMysqlConnection au tMysqlInput via un lien de type OnSubjobOk.
13. Connectez le composant **tMysqlInput** au **tMysqlCommit** à l’aide d’un lien **Trigger > OnSubjobOk**.

**Configurer les composants**

**Procédure**

1. Double-cliquez ensuite sur le composant **tMap** pour ouvrir l’éditeur graphique de mapping.
2. La table **Output** (qui s’est créée automatiquement quand vous avez lié le **tMap** au **tMysqlOutput**) sera composée des lignes concordantes du flux de référence (**large_data_volume**) et du flux principal (**people_age**).

Sélectionnez les lignes du flux principal que vous voulez passer en sortie et faites les glisser vers la table de sortie (dans le coin supérieur droit de l’éditeur de mapping) : ils y seront copiés.

Dans cet exemple, la sélection des lignes du flux principal inclut les champs suivants : *id*, *first_name*, *last_Name* et *age*.

Dans la table de référence, la colonne *city* est sélectionnée.

Faites glisser les colonnes sélectionnées des tables d’entrée (**people** et **large_volume**) vers la table de sortie.

3. Créez maintenant la jointure entre le flux principal et le flux de référence.

 Sélectionnez la colonne *age* dans la table du flux principal (en haut) et faites-la glisser vers la colonne *age* de la table du flux de référence (**large_volume** dans cet exemple).

L’icône d’une clé apparaît à côté de l’expression liée dans la table de référence. La jointure est maintenant établie.

4. Cliquez sur le bouton **tMap Settings** puis sur le bouton [...] correspondant à **Lookup Model** et sélectionnez l’option **Reload at each row** dans la boîte de dialogue [**Options**] pour que la base de référence soit chargée à nouveau pour chaque nouvelle ligne traitée.
5. De la même manière, paramétrez **Match Model** comme **All matches** dans la table de référence pour récupérer dans le flux de sortie toutes les occurrences de **age** qui correspondent.

6. Implémentez ensuite le filtre, établi à partir de la colonne **age**, dans la table de référence. Le champ **GlobalMapKey** est créé automatiquement lorsque vous sélectionnez l’option **Reload at each row**. Vous pouvez donc utiliser cette expression pour filtrer les données de référence de manière dynamique, pour que seules les informations pertinentes soient chargées avec le flux principal.

   Comme l’a souligné l’introduction de ce scénario, le flux de données principal contient uniquement les personnes dont l’âge est de 40 ou 60. Pour éviter de charger toutes les lignes de la table de référence, dont les âges sont différents de 40 et 60, vous pouvez utiliser la valeur âge du flux principal comme variable globale pour alimenter le filtre de référence.
7. Glissez la colonne *Age* de la table de flux principal vers le champ *Expr.* dans la table de référence.
8. Renseignez ensuite le nom de la variable dans le champ **globalMap Key**, en utilisant une expression. Dans cet exemple, l’expression est : "people.Age". Cliquez sur le bouton **OK** pour enregistrer les paramètres de mapping et retourner dans l’éditeur graphique.
9. Pour terminer l’implémentation du filtrage dynamique du flux de référence, vous devez maintenant ajouter une clause **WHERE** à la requête de la base de données d’entrée.

```sql
'SELECT
  large_data_volume."id",
  large_data_volume."First_Name",
  large_data_volume."Last_Name",
  large_data_volume."Age",
  large_data_volume."Address",
  large_data_volume."City",
  large_data_volume."US_State_ID"
FROM large_data_volume"
WHERE AGE = "[[Integer]globalMap.get("people.Age")].toString();"
```

10. À la fin du champ **Query**, juste après la commande **Select**, entrez la clause de type **WHERE** suivante :
WHERE AGE = '+(Integer)globalMap.get("people.Age")+''

11. Assurez-vous que le type correspond à la colonne utilisée comme variable. Dans cet exemple, la donnée Age est du type Integer. Veillez aussi à utiliser la variable de la manière dont vous l’avez paramétrée dans le champ globalMap key de l’éditeur de mapping.

12. Double-cliquez ensuite sur le composant tMysqlOutput et vérifiez que le schéma correspond aux paramètres de mapping.

13. Cochez la case Use an existing connection afin d’utiliser la connexion créée.

14. Spécifiez dans le champ Table le nom de la table cible.

Dans les listes Action on table et Action on data, sélectionnez les actions que vous souhaitez effectuer.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl+S afin de sauvegarder votre Job.
3. Dans la vue Debug Run, cliquez sur le bouton Traces Debug pour voir la progression du traitement des données.

Pour une meilleure visibilité, il est possible d’agrandir la vue du Job dans l’éditeur graphique lors de l’exécution en double-cliquant simplement sur l’onglet portant le nom du Job.
Les données de référence sont chargées à nouveau pour chaque ligne du flux principal qui correspond à la contrainte d’âge. Toutes les correspondances à la condition age sont récupérées à partir des lignes de la table de référence et regroupées dans le flux de sortie.

Si vous vérifiez les données contenues dans la table people_mixandmatch nouvellement créée, vous y trouverez toutes les copies basées sur age correspondant aux différentes personnes dont l’âge vaut 40 ou 60, ainsi que les villes où ils sont allés.

```
SELECT * FROM `mytalenddb`.`people_mixandmatch`;
```

Mapping avec des tables de sortie

Le scénario suivant décrit un Job traitant des flux de rejet sans les séparer du flux principal.
**Construire le Job**

**Procédure**

1. Dans la vue **Repository**, cliquez sur **Metadata > File delimited**. Glissez-déposez la métadonnée **customers** dans l'espace de modélisation graphique.
   
   La métadonnée **customers** contient des informations relatives aux clients, comme leur ID, leur nom, leur adresse, etc.
   
   Pour plus d'informations concernant la centralisation des métadonnées, consultez le **Guide utilisateur du Studio Talend**.

2. Une boîte de dialogue s'ouvre et vous demande de choisir le type de composant que vous souhaitez utiliser. Sélectionnez le composant **tFileInputDelimited** et cliquez sur **OK**.

3. Ensuite, glissez-déposez la métadonnée **states**. Sélectionnez le même composant dans la boîte de dialogue et cliquez sur **OK**.
   
   La métadonnée **states** contient l'identifiant de l'état, ainsi que son nom.

4. Glissez-déposez de la **Palette** dans l'espace de modélisation graphique un **tMap** et deux composants **tLogRow**.

5. Reliez le composant **customers** au **tMap** à l'aide d'un lien **Row > Main**.

6. Reliez le composant **states** au **tMap**, en utilisant un lien **Row > Main**. Il sera automatiquement défini comme **Lookup**.

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le composant **tMap** afin d’ouvrir le **Map Editor**.
   
   Glissez-déposez la colonne **idState** dans la première ligne de la seconde table, afin de créer une jointure.
   
   Cliquez sur le bouton **tMap settings** et sélectionnez **Inner Join** pour **Join Model**.

2. Cliquez sur le bouton **Property settings** en haut de la zone d’entrée afin d’ouvrir la boîte de dialogue [Property settings] et décochez la case **Die on error** si vous souhaitez gérer les erreurs d'exécution.
La table **ErrorReject** est maintenant créée.

3. Sélectionnez les colonnes $id$, $idState$, $RegTime$ et $RegisterTime$ de la table d'entrée et déposez-les dans la table **ErrorReject**.

4. Cliquez sur le bouton [+] en haut à droite de l'éditeur afin d'ajouter une table de sortie. Dans la boîte de dialogue qui s'ouvre, sélectionnez **New output**. Dans le champ à côté, saisissez le nom de la table, **out1**. Cliquez sur **OK**.

5. Déposez les colonnes suivantes des tables d'entrée dans la table **out1** : $id$, $CustomerName$, $idState$ et $LabelState$.

   Ajoutez deux colonnes $RegTime$ et $RegisterTime$ à la fin de la colonne **out1** et paramétrez leur format de date : "dd/MM/yyyy HH:mm" et "yyyy-MM-dd HH:mm:ss.SSS", respectivement.

6. Cliquez sur le champ **Expression** pour $RegTime$ et appuyez sur **Ctrl+Espace** pour afficher la liste d’autocomplétion. Sélectionnez `TalendDate.parseDate`. Changez le modèle en "dd/MM/yyyy HH:mm", `row1.RegTime`.


8. Cliquez sur le bouton [+] en haut à droite de l'éditeur, afin d'ajouter une table de sortie. Dans la boîte de dialogue qui s'ouvre, sélectionnez Create join table from, choisissez Out1 et donnez le nom rejectInner. Cliquez sur OK.

9. Cliquez sur le bouton tMap settings et sélectionnez true pour Catch lookup inner join reject, afin de gérer les rejets.

10. Déposez les colonnes id, CustomerName et idState des tables d'entrée dans les colonnes correspondantes de la table rejectInner.

   Cliquez dans le champ Expression de la colonne LabelState et saisissez "UNKNOWN".

11. Cliquez dans le champ Expression de la colonne RegTime, appuyez sur Ctrl+Espace, et sélectionnez TalendDate.parseDate. Changez le modèle en ("dd/MM/yyyy HH:mm", row1.RegTime.)

12. Dans la colonne RegisterTime, appuyez sur Ctrl+Espace, et sélectionnez TalendDate.parseDate, mais changez le modèle en ("yyyy-MM-dd HH:mm:ss.SSS", row1.RegisterTime).

   Si les données de la ligne row1 ont un modèle incorrect, elles seront retournées par le flux ErrorReject.
Cliquez sur OK afin de valider les modifications et fermer l’éditeur.

   Cliquez sur Sync columns pour récupérer la structure du schéma depuis le mapper si nécessaire.
   Dans la zone Mode, sélectionnez Table.
   Répétez l’opération avec le second tLogRow.

**Exécuter le Job**

**Procédure**

1. Appuyez sur Ctrl+S afin de sauvegarder le Job.

**Résultats**

La console de la vue Run affiche les flux de sortie principal et le flux ErrorReject. Le flux de sortie principal unifie les données valides et les rejets Inner Join, alors que le flux ErrorReject contient les informations des erreurs concernant les lignes ayant un format de date non analysable.
tMapRDBClose

Ce composant ferme une connexion à MapRDB que vous avez définie dans un même Job.

**Propriétés du tMapRDBClose Standard**

Ces propriétés sont utilisées pour configurer le tMapRDBClose s'exécutant dans le framework de Jobs Standard.

Le composant tMapRDBClose Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Basic settings**

| Component list | Sélectionnez le composant tMapRDBConnection dans la liste si plus d'une connexion est définie pour le Job utilisé. |

**Advanced settings**

| tStat Catcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| Règle d'utilisation | Ce composant doit être utilisé avec les composants MapRDB et plus particulièrement avec le tMapRDBConnection. |
Avant de commencer, vérifiez que tous les prérequis de l’IP de Loopback (rebouclage) attendus par votre base de données sont respectés.

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Scénario associé

Ce composant présente des similarités avec le tHBaseClose. Pour un scénario associé à l’utilisation du tHBaseClose, consultez Scénario : Echanger des données clients avec HBase à la page 1501.
tMapRDBConnection

Ce composant établit une connexion à MapRDB à réutiliser dans d’autres composants MapRDB au sein du Job.

Propriétés du tMapRDBConnection Standard

Ces propriétés sont utilisées pour configurer le tMapRDBConnection s’exécutant dans le framework de Jobs Standard.

Le composant tMapRDBConnection Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Property type | Peut être Built-In ou Repository.  
|               | Built-In : propriétés utilisées ponctuellement.  
|               | Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.  
|               | Les propriétés sont stockées centralement sous le nœud Hadoop Cluster de la vue Repository.  

| Distribution et Version | Sélectionnez la distribution MapR à utiliser. Seules les versions 5.2 et supérieures de MapR sont supportées par les composants MapRDB.  
|                         | Si la distribution que vous devez utiliser pour votre base de données MapRDB n’est pas officiellement supportée par ce composant MapRBD, c’est-à-dire, si la distribution de MapR ne s’affiche pas dans la liste déroulante Version de ce composant ou si cette distribution n’est pas MapR, sélectionnez Custom.  
|                         | 1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,  
|                         | 2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.  
|                         | Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import.  

from existing version, afin de se baser sur une distribution existante pour ajouter les jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

| Hadoop version of the distribution | Cette liste s’affiche uniquement lorsque vous avez sélectionné Custom dans la liste des distributions à connecter à un cluster n’étant pas officiellement supporté par le Studio. Dans cette situation, sélectionnez la version de Hadoop de ce cluster personnalisé, Hadoop 1 ou Hadoop 2. |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
Dans ce cas, les champs ou les options utilisée pour configurer la connexion Hadoop et/ou la sécurité Kerberos sont masqués.

Si vous souhaitez utiliser certains paramètres comme les paramètres Kerberos mais que ces paramètres ne sont pas inclus dans les fichiers de configuration Hadoop, vous devez créer un fichier appelé `talend-site.xml` et mettre ce fichier dans le répertoire défini dans `$HADOOP_CONF_DIR`. Le fichier `talend-site.xml` doit se présenter comme suit :

```xml
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>talend.kerberos.authentication</name>
    <value>kinit</value>
    <description>Set the Kerberos authentication method to use. Valid values are: kinit or keytab. </description>
  </property>
  <property>
    <name>talend.kerberos.keytab.principal</name>
    <value>user@BIGDATA.COM</value>
    <description>Set the keytab's principal name. </description>
  </property>
  <property>
    <name>talend.kerberos.keytab.path</name>
    <value>/kdc/user.keytab</value>
    <description>Set the keytab's path. </description>
  </property>
  <property>
    <name>talend.encryption</name>
    <value>none</value>
    <description>Set the encryption method to use. Valid values are: none or ssl. </description>
  </property>
  <property>
    <name>talend.ssl.trustStore.path</name>
    <value>ssl</value>
    <description>Set SSL trust store path. </description>
  </property>
  <property>
    <name>talend.ssl.trustStore.password</name>
    <value>ssl</value>
```
Les paramètres lus depuis ces fichiers de configuration écrasent ceux utilisés par défaut dans le Studio. Lorsqu’un paramètre n’existe pas dans ces fichiers de configuration, le paramètre par défaut est utilisé.

<table>
<thead>
<tr>
<th>Use kerberos authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si la base de données choisie utilise la sécurité Kerberos, cochez cette case puis saisissez le nom des principaux dans les champs affichés. Vous pouvez trouver cette information dans le fichier hbase-site.xml du cluster utilisé.</td>
</tr>
<tr>
<td>• Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.</td>
</tr>
<tr>
<td>Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.</td>
</tr>
<tr>
<td>Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case Use a keytab to authenticate. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.</td>
</tr>
<tr>
<td>Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser.</td>
</tr>
<tr>
<td>Pour plus d’informations relatives à la configuration de Kerberos pour votre base de données du cluster MapR, consultez Configuring Kerberos Authentication.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si vous devez utiliser une configuration personnalisée pour votre base de données, complétez ce tableau avec la ou les propriétés à personnaliser. Ensuite, lors de</td>
</tr>
</tbody>
</table>
l'exécution, la ou les propriétés écraseront celles qui ont été définies au préalable pour votre base de données.

Par exemple, vous devez définir la valeur de la propriété `dfs.replication` à 1 pour la configuration de la base de données. Vous devez ajouter par la suite une ligne à ce tableau en cliquant sur le bouton + et en saisissant le nom et la valeur de cette propriété dans cette ligne.

**tStatCatcher Statistics**

Cochez cette case afin de collecter les données du log au niveau du composant.

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

**Utilisation**

**Règle d’utilisation**

Ce composant est généralement utilisé avec d’autres composants MapRDB, notamment le **tMapRDBClos**.

**Prérequis**

Avant de commencer, vérifiez que tous les prérequis de l’IP de Loopback (rebouclage) attendus par votre base de données sont respectés.

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le **Studio Talend**. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le **Studio Talend** et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est `lib\MapRClient.dll` dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).
Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante: no MapRCient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Scénario associé

Ce composant est similaire au tHBaseConnection. Pour un scénario relatif à l’utilisation du tHBaseConnection, consultez Scénario : Echanger des données clients avec HBase à la page 1501.
tMapRDBInput

Ce composant lit les données d’une base de données MapRDB avant d’extraire les colonnes sélectionnées.

Le tMapRDBInput extrait les colonnes correspondantes vers la définition du schéma avant de faire passer ces colonnes vers le nouveau composant via le lien Row > Main.

Propriétés du tMapRDBInput Standard

Ces propriétés sont utilisées pour configurer le tMapRDBInput s’exécutant dans le framework de Jobs Standard.

Le composant tMapRDBInput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Property type | Peut être **Built-In** ou **Repository**.  
**Built-In** : propriétés utilisées ponctuellement.  
**Repository** : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.  
Les propriétés sont stockées centralement sous le nœud Hadoop Cluster de la vue Repository.  

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  

| Distribution et Version | Sélectionnez la distribution MapR à utiliser. Seules les versions 5.2 et supérieures de MapR sont supportées par les composants MapRDB.  
Si la distribution que vous devez utiliser pour votre base de données MapRDB n’est pas officiellement supportée par ce composant MapRBD, c’est-à-dire, si la distribution de MapR ne s’affiche pas dans la liste déroulante Version de ce composant ou si cette distribution n’est pas MapR, sélectionnez Custom.  
1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,  
2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.  
Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip.
de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l'évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d'utiliser l'option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l'opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d'Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Version Hadoop de la distribution</th>
<th>Cette liste s’affiche uniquement lorsque vous avez sélectionné Custom dans la liste des distributions à connecter à un cluster n’étant pas officiellement supporté par le Studio. Dans cette situation, sélectionnez la version de Hadoop de ce cluster personnalisé, Hadoop 1 ou Hadoop 2.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zookeeper quorum</td>
<td>Saisissez le nom ou l’URL du service Zookeeper utilisé pour coordonner les transactions entre votre Studio et votre base de données. Notez que, lorsque vous configurez Zookeeper, vous pouvez avoir besoin de configurer explicitement la propriété zookeeper.znode.parent pour définir le chemin vers le nœud znode racine contenant tous les znodes créés et utilisés par votre base de données. Cochez la case la case Set Zookeeper znode parent afin de définir cette propriété.</td>
</tr>
<tr>
<td>Zookeeper client port</td>
<td>Saisissez le numéro du port d’écoute client du service Zookeeper que vous utilisez.</td>
</tr>
<tr>
<td>Use kerberos authentication</td>
<td>Si la base de données choisie utilise la sécurité Kerberos, cochez cette case puis saisissez le nom des principaux dans les champs affichés. Vous pouvez</td>
</tr>
</tbody>
</table>
trouver cette information dans le fichier `hbase-site.xml` du cluster utilisé.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases Force MapR ticket authentication et Use Kerberos authentication. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case Use a keytab to authenticate. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est `user1` et le principal à utiliser est `guest`. Dans cette situation, assurez-vous que `user1` a les droits de lecture pour le fichier keytab à utiliser.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métdonnée du schéma dans la fenêtre [Repository Content].
### Built-In
Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository
Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

### Set table Namespace mappings
Saisissez la chaîne de caractères à utiliser pour construire le mapping entre une table Apache HBase et une table MapR.

Pour plus d’informations concernant la syntaxe valide à utiliser, consultez [http://doc.mapr.com/display/MapR40x/Mapping+Table+Namespace+Between+Apache+HBase+Tables+and+MapR+Tables](http://doc.mapr.com/display/MapR40x/Mapping+Table+Namespace+Between+Apache+HBase+Tables+and+MapR+Tables) (en anglais).

### Table name
Saisissez le nom de la table de la base de données de laquelle vous souhaitez extraire les colonnes.

### Define a row selection
Cochez cette case et, dans les champs Start row et End row, saisissez les Row Keys correspondants afin de spécifier la plage de lignes que vous souhaitez extraire par le composant.

À la différence des filtres que vous pouvez définir à l'aide de l'option **Is by filter** nécessitant le chargement de tous les enregistrements avant de pouvoir filtrer les lignes à utiliser, cette fonctionnalité vous permet de sélectionner directement les lignes qui vous intéressent.

### Mapping
Renseignez cette table afin de mapper les colonnes de la table à utiliser avec les colonnes du schéma défini pour le flux de données à traiter.

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau de chaque composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Si vous devez utiliser la configuration personnalisée pour votre base de données, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) vont écraser les propriétés utilisées par le Studio Talend. Par exemple, vous devez définir la valeur de la propriété dfs.replication à 1 pour la configuration de base de données. Vous devez ensuite ajouter une ligne à la table à l’aide du bouton [+] et saisir le nom et la valeur de cette propriété dans la ligne. <strong>Remarque :</strong> Cette table n’est pas disponible quand vous utilisez une connexion existante en cochant la case <strong>Using an existing connection</strong> dans la vue Basic settings.</td>
</tr>
<tr>
<td><strong>Is by filter</strong></td>
<td>Cochez cette case pour utiliser des filtres HBase afin d'effectuer une sélection granulaire fine depuis votre base de données, comme une sélection de clés ou de valeurs, selon des expressions régulières. Une fois cette case cochée, la table <strong>Filter</strong> utilisée pour définir les conditions de filtre devient disponible. Cette fonctionnalité tire parti des filtres fournis par HBase et sujets aux contraintes présentées dans la documentation Apache HBase. Un niveau avancé de connaissances de HBase est requis pour que les utilisateurs puissent utiliser pleinement ces filtres.</td>
</tr>
</tbody>
</table>
| **Logical operation** | Sélectionnez l'opérateur à utiliser pour définir la relation logique entre les filtres. Les opérateurs disponibles sont :
- **And** : chaque condition de filtre définie doit être satisfaite. Elle représente la relation **FilterList.Operator.MUST_PASS_ALL**.
- **Or** : au moins une des conditions de filtre définie doit être satisfaite. Elle représente la relation : **FilterList.Operator.MUST_PASS_ONE**. |
| **Filter** | Cliquez sur le bouton sous cette table pour ajouter autant de lignes que nécessaire. Chaque ligne représente un filtre. Les paramètres que vous devez configurer pour un filtre sont :
- **Filter type** : la liste déroulante présente les types de filtre prédéfinis par HBase. Sélectionnez le type de filtre à utiliser.
- **Filter column** : saisissez le nom de la colonne (qualifier) sur laquelle vous devez appliquer le filtre actif. Ce paramètre est obligatoire selon le type de filtre et le comparateur que vous utilisez. Par exemple, ce paramètre n'est pas utilisé par le type **Row Filter** mais est requis pour le type **Single Column Value Filter**.
- **Filter family** : saisissez la famille de colonne sur laquelle vous devez appliquer le filtre actif. Ce paramètre est obligatoire selon le type de filtre et le comparateur que vous utilisez. Par exemple, ce paramètre n'est pas utilisé par le type **Row Filter** mais est requis pour le type **Single Column Value Filter**.
- **Filter operation** : sélectionnez dans la liste déroulante l'opération à utiliser pour le filtre actif.
- **Filter Value** : saisissez le valeur sur laquelle utiliser l'opérateur sélectionné dans la liste **Filter operation**.
- **Filter comparator type** : sélectionnez le type de comparateur à combiner au filtre que vous utilisez. Selon le type de filtre (**Filter type**) que vous utilisez, certains paramètres, voire tous les paramètres sont obligatoires. Pour plus d’informations, consultez Filtres HBase à la page 1493. |
Global Variables


Utilisation

| Règle d’utilisation | Ce composant est composant de début d’un Job et requiert toujours un lien de sortie. |

| Prérequis | Avant de commencer, vérifiez que tous les prérequis de l’IP de Loopback (rebouclage) attendus par votre base de données sont respectés. La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR. • Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path. • Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque |
native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données **(Data viewer)** afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

**Scénario associé**

Ce composant présente des similarités avec le composant **tHBaseInput**. Pour un scénario associé au **tHBaseInput**, consultez **Scénario : Echanger des données clients avec HBase** à la page 1501.
**tMapRDBOutput**

Ce composant écrit les colonnes de données dans une base de données MapRDB.

Le tMapRDBOutput reçoit des données issues du composant précédent, créé la table dans une base de données MapRDB et écrit les données reçues dans cette table.

**Propriétés du tMapRDBOutput Standard**

Ces propriétés sont utilisées pour configurer le tMapRDBOutput s’exécutant dans le framework de Jobs Standard.

Le composant tMapRDBOutput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Property type       | Peut être **Built-In** ou **Repository**.
|---------------------|--------------------------------------------------
|                     | **Built-In** : propriétés utilisées ponctuellement.  
|                     | **Repository** : sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |
| Distribution et Version | Sélectionnez la distribution MapR à utiliser. Seules les versions 5.2 et supérieures de MapR sont supportées par les composants MapRDB.  
|                     | Si la distribution que vous devez utiliser pour votre base de données MapRDB n’est pas officiellement supportée par ce composant MapRBD, c’est-à-dire, si la distribution de MapR ne s’affiche pas dans la liste déroulante **Version** de ce composant ou si cette distribution n’est pas MapR, sélectionnez **Custom**.  
|                     | 1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,  
|                     | 2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.  
|                     | Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre |
connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Noter que certaines versions personnalisées ne sont pas officiellement supportées par **Talend**. **Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

### Hadoop version of the distribution
Cette liste s’affiche uniquement lorsque vous avez sélectionné **Custom** dans la liste des distributions à connecter à un cluster n’étant pas officiellement supporté par le Studio. Dans cette situation, sélectionnez la version de Hadoop de ce cluster personnalisé, **Hadoop 1** ou **Hadoop 2**.

### Zookeeper quorum
Saisissez le nom ou l’URL du service Zookeeper utilisé pour coordonner les transactions entre votre Studio et votre base de données. Notez que, lorsque vous configurez Zookeeper, vous pouvez avoir besoin de configurer explicitement la propriété `zookeeper.znode.parent` pour définir le chemin vers le nœud znode racine contenant tous les znodes créés et utilisés par votre base de données. Cochez la case **Set Zookeeper znode parent** afin de définir cette propriété.

### Zookeeper client port
Saisissez le numéro du port d’écoute client du service Zookeeper que vous utilisez.

### Use kerberos authentication
Si la base de données choisie utilise la sécurité Kerberos, cochez cette case puis saisissez le nom des principaux dans les champs affichés. Vous pouvez trouver cette information dans le fichier `hbase-site.xml` du cluster utilisé.
• Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case **Use a keytab to authenticate**. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est **user1** et le principal à utiliser est **guest**. Dans cette situation, assurez-vous que **user1** a les droits de lecture pour le fichier keytab à utiliser.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.
**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

**Set table Namespace mappings**

Saisissez la chaîne de caractères à utiliser pour construire le mapping entre une table Apache HBase table et une table MapR.

Pour plus d’informations concernant la syntaxe valide à utiliser, consultez http://doc.mapr.com/display/MapR40x/Mapping+Table+Namespace+Between+Apache+HBase+Tables+and+MapR+Tables (en anglais).

**Table name**

Saisissez la table du HBase que vous devez créer.

**Action on table**

Sélectionnez l’action dont vous avez besoin pour créer une table.

**Custom Row Key**

Cochez la case afin d’utiliser les clés de ligne personnalisées. Lorsque la case est cochée, le champ correspondant s’affiche. Saisissez ensuite les lignes de clé personnalisées pour indexer les lignes de la table qui est créée.

Par exemple, vous pouvez saisir "France"+Numeric.sequence("s1",1,1) afin de produire des séries de clé de lignes : France1, France2, France3, etc.

**Families**

Complétez cette table pour spécifier les colonnes à créer ou les familles de colonne auxquelles elles appartiennent respectivement. La colonne Column de cette table est automatiquement renseignée lorsque vous avez défini le schéma.

**Die on error**

Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

**Advanced settings**

**Use batch mode**

Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.

**Batch size**

Spécifiez le nombre d’enregistrements à traiter dans chaque lot.
Ce champ est disponible uniquement lorsque la case **Use batch mode** est cochée.

### Properties

Si vous devez utiliser la configuration personnalisée pour votre base de données, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) vont écraser les propriétés utilisées par le Studio Talend.

Par exemple, vous devez définir la valeur de la propriété *dfs.replication* à 1 pour la configuration de base de données. Vous devez ensuite ajouter une ligne à la table à l’aide du bouton [*] et saisir le nom et la valeur de cette propriété dans la ligne.

**Remarque :**

Cette table n’est pas visible lorsque vous utilisez une connexion existante cochant la case **Using an existing connection** dans la vue **Basic settings**.

### tStatCatcher Statistics

Cochez la case afin de collecter les données de log au niveau de chaque composant.

### Family parameters

Saisissez les noms et, si nécessaire, les options de performance personnalisées des familles de colonne que vous allez créer. Cette fonctionnalité tire parti des attributs définis par le modèle de données HBase. Pour plus d’informations sur ces options, consultez la documentation Apache HBase.

**Remarque :**

Le paramètre *Compression type* vous permet de sélectionner le format de compression des données de sortie.

### Global Variables

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.
## Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est un composant de fin et requiert toujours un lien d’entrée.</th>
</tr>
</thead>
</table>
| Prérequis           | Avant de commencer, vérifiez que tous les prérequis de l’IP de Loopback (rebouclage) attendus par votre base de données sont respectés.  
La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend.  
La liste suivante présente des informations d’exemple relatives à MapR.  
   • Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les bibliothèques du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).  
Si vous n’ajoutez pas de bibliothèque, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.  
   • Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.  
Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez. |

## Scénario associé

Ce composant présente des similarités avec le tHBaseOutput. Pour un scénario associé au tHBaseOutput, consultez Scénario : Echanger des données clients avec HBase à la page 1501.
tMapROjaiOutput

Ce composant insère, met à jour ou supprime des documents dans une base de données MapR-DB utilisée en tant que base de données de documents, en se basant sur le flux entrant du composant précédent dans le Job.

Vous devez avoir installé et configuré votre client MapR avant d’utiliser ce composant. Un fichier \opt\mapr\conf\mapr-clusters.conf est automatiquement généré à partir du processus de configuration du Client, afin de stocker les informations de connexion à votre cluster MapR et à votre base de données MapR-DB. Lors de l’exécution, le tMapROjaiOutput lit ces informations de connexion pour se connecter à la base de données MapR-DB à utiliser.

Pour plus d’informations concernant l’installation et la configuration de votre client MapR, consultez la page Setting up the Client (en anglais) de la documentation MapR.

Propriétés du tMapROjaiOutput Standard

Ces propriétés sont utilisées pour configurer le tMapROjaiOutput s’exécutant dans le framework de Jobs Standard.

Le composant tMapROjaiOutput Standard appartient à la famille Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table</th>
<th>Saisissez le nom de la table à traiter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB versions</td>
<td>Sélectionnez la version de votre base de données.</td>
</tr>
</tbody>
</table>

| Use kerberos authentication                                                  | Si vous accédez à une base de données MapR-DB de type OJAI s’exécutant avec une sécurité Kerberos, cochez cette case, saisissez le nom et le mot de passe du Principal, dans les champs qui s’affichent. |
|                                                                              | À chaque démarrage du Job, le composant envoie ses informations d’authentification à Kerberos pour un nouveau ticket kinit. |
|                                                                              | • Si le mécanisme de sécurité par ticket MapR est également configuré avec Kerberos, ce composant émet une commande mapr login pour obtenir un ticket MapR en même temps. |
|                                                                              | • Si le mécanisme de sécurité par ticket MapR est configuré pour votre cluster alors que Kerberos ne l’est pas, laissez cette case décochée. Le composant obtient un ticket MapR via votre client MapR Client. |
### Use a keytab to authenticate

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l'aide d'un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est *user1* et le principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits de lecture pour le fichier keytab à utiliser.

### Action on table

Sélectionnez une opération à effectuer sur la table définie, vous pouvez effectuer l’une des opérations suivantes :

- **None** : aucune opération n’est effectuée.
- **Drop and create table** : la table est supprimée et créée à nouveau.
- **Create table** : la table n’existe pas et est créée.
- **Create table if does not exist** : la table est créée si elle n’existe pas.
- **Drop table if exist and create** : la table est supprimée si elle existe déjà et créée à nouveau.
- **Truncate** : le contenu de la table est supprimé.

### Action on data

Sélectionnez une action à effectuer sur les données de la table définie.

- **Insert** : ajoute de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
- **Replace** : si la table contient déjà des données, supprime toutes les données existantes et insère les nouvelles données. Si la table est vide, insère les nouvelles données.
- **Insert or Replace** : regarde les ID des documents, remplace les documents dont l’ID existe dans la base de données et dans les données à écrire et insère les documents dont l’ID n’existe pas dans la base de données.
- **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas dans le pool d’index, un nouvel enregistrement est inséré.
- **Delete** : supprime les entrées correspondantes au flux d’entrée.
## Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réserve *line* lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

Cliquez sur **Sync columns** afin de récupérer le schéma du composant précédent relié dans le Job.

### Built-In

Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

### Repository

Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Bulk write

Cochez cette case pour insérer, mettre à jour ou supprimer des données en masse.

Dans le champ **Bulk write size**, saisissez la taille de chaque groupe de requêtes à traiter avec MapR-DB.

### Mapping

Chaque colonne du schéma défini pour ce composant représente un champ des documents à lire. Dans cette table, vous devez spécifier les nœuds parents de ces champs, s’il y en a.

Par exemple, dans le document se présentant comme suit :

```json
{
    _id: ObjectId("5099803df3f4948bd2f98391"),
    ...}
```
Les champs `first` et `last` ont un nœud père `person` mais le champ `_id` ne contient aucun nœud père. Cela fait, la table `Mapping` doit ressembler à ceci :

<table>
<thead>
<tr>
<th>Column</th>
<th>Parent node path</th>
</tr>
</thead>
<tbody>
<tr>
<td>_id</td>
<td>&quot;person&quot;</td>
</tr>
<tr>
<td>first</td>
<td>&quot;person&quot;</td>
</tr>
<tr>
<td>last</td>
<td>&quot;person&quot;</td>
</tr>
</tbody>
</table>

**Die on error**

Cette case est décochée par défaut, pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Variables globales**

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

- **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable `After` et retourne un entier.

- **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable `After` et retourne un entier.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Usage**

**Règle d’utilisation**

Le tMapROjaiOutput exécute l’action définie sur les documents dans une base de données MapR-DB, en se basant sur le flux entrant provenant du composant précédent dans votre Job.
Écrire des données relatives à des candidats dans une base de données MapR-DB OJAI

Ce scénario utilise un composant tMapROjaiOutput pour écrire des données concernant des candidats dans une base de données MapR-DB OJAI.

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

- Le tFixedFlowInput fournit l’échantillon de données à écrire dans la base de données.
- Le tMapROjaiOutput écrit l’échantillon de données dans la base de données.

Lire l’échantillon de données des candidats

**Procédure**

1. Assurez-vous que le client MapR a bien été installé et configuré sur la machine sur laquelle s’exécute le Job courant.
2. Déposez un tFixedFlowInput et un tMapROjaiOutput dans l’espace de modélisation graphique.
3. Reliez le tFixedFlowInput au tMapROjaiOutput à l’aide d’un lien Row > Main et acceptez la propagation du schéma du tMapROjaiOutput.
5. Cliquez sur le bouton [...] pour ouvrir l’éditeur de schéma.
La colonne `_id` existe déjà, elle a été récupérée du `tMapROjaiOutput` au cours des étapes précédentes, pour fournir les ID techniques des documents à stocker dans une base de données MapR Ojai. Cette colonne est requise par le `tMapROjaiOutput`.

Cliquez sur le bouton `[+]` pour ajouter les autres colonnes, que vous allez renommer `firstname`, `lastname`, `sex`, `married`, `age` et `salary`, respectivement. Le type de la colonne `married` doit être `Boolean`, le type de la colonne `age` doit être `Integer` et le type de la colonne `salary` doit être `Float`.


7. Dans la zone Mode, sélectionnez le bouton radio Use Inline table, pour afficher la table dans laquelle vous avez ajouté l’échantillon de données.

8. Cliquez sur le bouton `[+]` pour ajouter quatre lignes et saisissez le jeu de données comme dans la capture d’écran.

Cet échantillon de données est utilisé à des fins de démonstration.

Les données de type `String` doivent être entourées de guillemets doubles et les données des autres types ne doivent pas contenir de guillemets.

**Écrire l’échantillon de données dans la base de données MapR OJAI**

**Procédure**

1. Double-cliquez sur `tMapROjaiOutput` pour ouvrir sa vue Component.
2. Dans le champ **Table**, saisissez le nom de la table à utiliser dans votre base de données MapR OJAI.

3. Dans la liste **DB Version**, sélectionnez la version de la base de données MapR OJAI que vous utilisez.

Lors de l’exécution, le tMapROjaiOutput lit automatiquement les informations de connexion à la base de données depuis le fichier `\opt\mapr\conf\mapr-clusters.conf` de votre client MapR. Dans cet exemple, le contenu du fichier se présente comme suit :

```
mapr521   mapr521:7222
```

4. Dans la liste **Action on table**, sélectionnez **Drop table if exists and create** et, dans la liste **Action on data**, sélectionnez **Insert**.

5. Dans la table **Mapping**, les colonnes du schéma s’affichent automatiquement dans la colonne **Column**. Dans la colonne **Parent field path**, saisissez "Name" entre guillemets doubles dans la ligne `firstname` et la liste `lastname` et saisissez "Status" entre guillemets doubles dans la ligne `sex`, dans la ligne `married` et dans la ligne `age`.

Cette configuration regroupe les noms et prénoms des candidats dans le champ **Name** et leur genre, leur statut marital et leur âge, dans le champ **Status** dans le flux de données écrit en sortie.

6. Appuyez sur **F6** pour exécuter le Job.
**tMapRStreamsCommit**

Ce composant se connecte à un tMapRStreamsInput donné pour effectuer un commit d’un offset consommateur.

Le tMapRStreamsCommit sauvegarde l’état courant du tMapRStreamsInput auquel il était connecté. Notez que le terme “commit” dans ce composant signifie sauvegarder les messages consommés par ce tMapRStreamsInput au moment du commit.

**Propriétés du tMapRStreamsCommit Standard**

Ces propriétés sont utilisées pour configurer le tMapRStreamsCommit s’exécutant dans le framework de Jobs Standard.

Le composant tMapRStreamsCommit Standard appartient à la famille Internet.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Commit offsets from | Sélectionnez le composant tMapRStreamsInput duquel les messages consommés sont commités. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour rassembler les métadonnées de traitement au niveau du Job ainsi qu’au niveau de chaque composant. |

**Utilisation**

| Règle d’utilisation | Même si le tMapRStreamsCommit peut être utilisé comme composant de fin dans un flux de sous-job ou peut être appelé indépendamment comme sous-job séparé, il est généralement utilisé en standalone pour commiter un batch d’offsets en une fois. Si vous devez commiter régulièrement les offsets, il est recommandé d’utiliser la fonctionnalité d’Auto-commit dans le tMapRStreamsInput. |

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tMapRStreamsConnection

Ce composant ouvre une connexion réutilisable à un cluster MapR Streams donné, afin que les autres composants MapR Streams puissent réutiliser cette connexion.

**Propriétés du tMapRStreamsConnection Standard**

Ces propriétés sont utilisées pour configurer le tMapRStreamsConnection s’exécutant dans le framework de Jobs Standard.

Le composant tMapRStreamsConnection Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
</tr>
</thead>
</table>
| **Property type** | Peut être **Built-In** ou **Repository**.  
**Built-In** : propriétés utilisées ponctuellement.  
**Repository** : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.  
Les propriétés sont stockées centralement sous le nœud Hadoop Cluster de la vue Repository. |

<table>
<thead>
<tr>
<th>Distribution et Version</th>
</tr>
</thead>
</table>
| **Distribution et Version** | Sélectionnez la distribution MapR à utiliser. Seules les versions 5.2 et supérieures de MapR sont supportées par les composants MapRDB.  
Si la distribution que vous devez utiliser pour votre base de données MapRDB n’est pas officiellement supportée par ce composant MapRDB, c’est-à-dire, si la distribution de MapR ne s’affiche pas dans la liste déroulante **Version** de ce composant ou si cette distribution n’est pas MapR, sélectionnez **Custom**.  
1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,  
2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.  
Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import**. |
from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

**NameNode URI**

Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée *masternode* comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un TLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

**Force MapR ticket authentication**

Si le cluster MapR utilisé est sécurisé avec le mécanisme d’authentification par ticket MapR, configurez l’authentification par ticket MapR en suivant l’explication présentée dans Configurer l’authentification par ticket à MapR à la page 1746.

Par ailleurs, si votre cluster utilise également Kerberos pour sécuriser vos streams MapR, cochez la case Use Kerberos authentication afin de configurer Kerberos pour votre Job.

- Vous pouvez également trouver un exemple de configuration de Kerberos pour un Job Talend Job dans Talend Help Center (https://help.talend.com). Bien que cet exemple utilise Cloudera à des fins de démonstration, les opérations décrites sont génériques et sont donc également applicables à MapR.
**Advanced settings**

| Hadoop properties | Le Studio Talend utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.
|                  | • Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste Property type, vous passiez de Repository à Built-in. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau de chaque composant. |

**Variables globales**

|                   | Ce composant est généralement utilisé avec d’autres composants MapR Streams. |

**Utilisation**

| Règle d’utilisation | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.
|                    | • Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib |

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : `no MapRClient in java.library.path`.

- Configurez l’argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]** dans le menu **Window**. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

---

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tMapRStreamsCreateStream

Ce composant crée un flux ou un topic MapR Streams que les autres composants MapR Streams peuvent utiliser.

Ce composant vous permet d'exécuter visuellement une commande pour créer ou mettre à jour un flux ou un topic.

**Propriétés du tMapRStreamsCreateStream Standard**

Ces propriétés sont utilisées pour configurer le tMapRStreamsCreateStream s'exécutant dans le framework de Jobs Standard.

Le composant tMapRStreamsCreateStream Standard appartient à la famille Internet.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Property type | Built-In : propriétés utilisées ponctuellement.  
|               | Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.  
|               | Les propriétés sont stockées centralement sous le nœud Hadoop Cluster de la vue Repository. |

| tMapRStreamsConnection | Sélectionnez le composant tMapRStreamsConnection afin de réutiliser les informations de connexion précédemment définies. Ce paramètre est obligatoire pour que ce composant se connecter à MapR Streams. |

| Action | Sélectionnez comment un flux ou un topic est créé. Les paramètres associés à chaque action sont affichés uniquement lorsque l’action en question est sélectionnée. |
|        | • Create stream : crée un flux. Si ce flux existe déjà, le Job complet est arrêté. |
|        | • Create stream if not exists : crée un flux lorsque le flux n’existe pas. Si ce flux existe déjà, le Job ignore la création pour passer à l’étape suivante. |
|        | • Alter stream : modifie la configuration d’un flux. |
|        | • Create topic : crée un topic. Si ce topic existe déjà, le Job complet est arrêté.  
|        | Notez que le flux auquel le topic appartient doit déjà exister. |
|        | • Create topic if not exists : crée un topic lorsque ce dernier n’existe pas. Si ce topic existe, le Job ignore la création pour passer à l’étape suivante.  
|        | Notez que le flux auquel ce topic appartient doit déjà exister. |
### Stream path

Saisissez le chemin d'accès au flux à créer ou mettre à jour dans MapR-FS.

### Stream permissions

Accordez les autorisations à un compte utilisateur sur le flux à créer ou mettre à jour. Ces autorisations concernent :

- Adminperm
- Consumeperm
- Copyperm
- Produceperm
- Topicperm

La syntaxe que vous devez utiliser est `u:username_or_userid_of_the_account`, entre guillemets doubles.

Lorsque vous avez sélectionné Alter stream dans la liste Action, plusieurs cases s’affichent dans la zone Stream permissions, afin que vous puissiez cocher la case correspondant à chaque autorisation à modifier et que vous accordiez cette autorisation à un utilisateur différent.

Pour plus d’informations concernant ces autorisations, consultez la documentation de MapR à l’adresse suivante MapR Streams Security (en anglais).

### Stream settings

Définissez la configuration du flux à créer ou mettre à jour. Les paramètres à configurer sont les suivants :

- **Automatic topic creation** : cochez la case Enable automatic topic creation pour permettre à un topic d’être automatiquement créé lorsqu’un producteur crée le premier message pour celui-ci et que ce topic n’existe pas déjà.

  Pour plus d’informations concernant cette création automatique, consultez la documentation de MapR à l’adresse Topic Creation (en anglais).

- **Default number of partitions** : ce paramètre définit le nombre de partitions créées dans chaque nouveau topic dans un flux.

  Pour plus d’informations concernant le nombre par défaut de partitions, consultez la documentation de MapR à l’adresse suivante Default Partitions (en anglais).

- **Message time to live** : ce paramètre définit la durée (en secondes) de la persistance des messages dans un flux.

  Pour plus d’informations concernant la durée de persistance des messages, consultez la documentation de MapR à l’adresse Time-to-live for Messages (en anglais).

- **Compression** : type des algorithmes de compression que MapR fournit pour compresser les fichiers stockés dans le cluster MapR.
Pour plus d’informations concernant la compression dans MapR, consultez la documentation de MapR à l’adresse Compression (en anglais).

Lorsque vous avez sélectionné l’option Alter stream dans la liste Action, plusieurs cases s’affichent dans la zone Stream settings, afin que vous puissiez cocher la case correspondant à chaque paramètre à modifier, puis mettre à jour ce dernier.

### Topic name
Saisissez le nom du topic à créer.
Ce champ s’affiche lorsque vous avez sélectionné l’une des actions relatives aux topics dans la liste Action.

### Number of partitions ou Set number of partitions
Saisissez le nombre de partitions à créer pour ce topic.
Ce champ s’affiche lorsque vous avez sélectionné l’une des actions relatives aux topics dans la liste Action.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour rassembler les métadonnées de traitement au niveau Job ainsi qu’au niveau de chaque composant. |

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé pour créer ou mettre à jour un topic ou un flux à utiliser par d’autres composants MapR Streams. |


### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tMapRStreamsInput

Ce composant transmet des messages au Job exécutant des transformations sur ces messages. Ce composant supporte uniquement les versions 5.2 et suivantes de MapR.

Propriétés du tMapRStreamsInput Standard

Ces propriétés sont utilisées pour configurer le tMapRStreamsInput s'exécutant dans le framework de Jobs Standard.

Le composant tMapRStreamsInput Standard appartient à la famille Internet.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Notez que le schéma de ce composant est en lecture seule. Il stocke les messages envoyés depuis le producteur des messages. |
|----------------------|---------------------------------------------------------------------------------------------------------|
| Output type          | Sélectionnez le type de données à envoyer au composant suivant.

Généralement, il est recommandé d’utiliser le type `String`, car le `tMapRStreamsInput` peut automatiquement traduire les messages MapR Streams de type byte[] en chaînes de caractères, afin de les traiter par le Job. Cependant, si le format des messages MapR Streams est inconnu du `tMapRStreamsInput`, comme Protobuf, vous pouvez sélectionner `byte[]` et utiliser un composant `Custom code`, comme le `tJavaRow`, afin de désérialiser les messages en chaînes de caractères, afin que les autres composants du Job puissent traiter ces messages. |
| Use an existing connection | Cochez cette case et, dans la liste qui s’affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies. |
| Distribution et Version | Sélectionnez la distribution MapR à utiliser. Seules les versions 5.2 et supérieures de MapR sont supportées par les composants MapRDB.

Si la distribution que vous devez utiliser pour votre base de données MapRDB n’est pas officiellement supportée par ce composant MapRBD, c’est-à-dire, si la distribution de MapR ne s’affiche pas dans la liste déroulante `Version` de ce composant ou si cette distribution n’est pas MapR, sélectionnez `Custom`. |
1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend**. **Talend** et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

<table>
<thead>
<tr>
<th>Topic name</th>
<th>Saisissez le nom du topic duquel le tMapRStreamsInput reçoit le flux des messages. Vous devez saisir le nom du flux auquel ce topic appartient. La syntaxe est <code>path_to_the_stream:topic_name</code>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer group ID</td>
<td>Saisissez le nom du groupe de consommateurs auquel vous souhaitez que le consommateur courant (le composant tMapRStreamsInput) appartienne. Ce groupe de consommateurs sera créé lors de l’exécution s’il n’existe pas.</td>
</tr>
<tr>
<td><strong>Reset offsets on consumer group</strong></td>
<td>Cochez cette case pour supprimer les offsets sauvegardés pour le groupe de consommateurs à utiliser, afin que ce groupe de consommateurs soit géré comme un nouveau groupe n'ayant consommé aucun message.</td>
</tr>
</tbody>
</table>
| **New consumer group starts from** | Sélectionnez le point de départ à partir duquel les messages d'un topic seront consommés.  
Dans MapR Streams, le numéro d'ID séquentiel d'un message se nomme offset. Lorsqu'un nouveau groupe de consommateurs démarre, dans cette liste, vous pouvez sélectionner beginning pour commencer la consommation à partir du composant le plus ancien du topic, ou sélectionner latest pour attendre un nouveau message.  
Notez que le groupe de consommateurs prend en compte uniquement les messages dont l’offset a été commité comme point de départ.  
Chaque groupe de consommateurs possède son propre compteur pour se rappeler la position d’un message consommé. Pour cette raison, une fois qu’un groupe de consommateurs a commencé à consommer des messages d’un sujet donné, un groupe de consommateurs reconnaît le message le plus récent en voyant simplement la position où son groupe a arrêté la consommation, plutôt que le sujet complet. Partant de ce principe, les comportements suivants peuvent être attendus :  
• Si vous reprenez un groupe de consommateurs existant, cette option détermine le point de départ de ce groupe de consommateurs, uniquement s’il n’a pas encore de point de départ commité. Sinon, ce groupe de consommateurs démarre du point de départ commité. Par exemple, un topic contient 100 messages. Si un groupe de consommateurs existants a bien traité 50 messages et a commité leurs offsets, le même groupe de consommateurs redémarre de l'offset 51.  
• si vous créez un nouveau groupe de consommateurs ou en réinitialisez un existant, ce qui signifie que ce groupe n’a consommé aucun message de ce topic et, lorsque vous le démarrez depuis le dernier message, ce nouveau groupe démarre et attend l’offset 101. |
| **Auto-commit offsets** | Cochez cette case pour que le tMapRStreamsInput sauvegarde automatiquement l’état de sa consommation, à la fin de chaque intervalle de temps donné. Vous devez définir cet intervalle dans le champ Interval qui s’affiche.  
Notez que les offsets sont commités uniquement à la fin de chaque intervalle. Si votre Job s’arrête au milieu d’un intervalle, l’état de consommation du message dans cet intervalle n’est pas commité. |
Stop after a maximum total duration (ms)

Cochez cette case et, dans le champ qui s’affiche, saisissez la durée (en millisecondes) à la fin de laquelle le tMapRStreamsInput arrête son exécution.

Stop after receiving a maximum number of messages

Cochez cette case et, dans le champ qui s’affiche, saisissez le nombre maximal de messages que le tMapRStreamsInput doit recevoir avant qu’il arrête automatiquement de s’exécuter.

Stop after maximum time waiting between messages (ms)

Cochez cette case et, dans le champ qui s’affiche, saisissez le temps d’attente (en millisecondes) durant lequel le tMapRStreamsInput attend un nouveau message. Si le tMapRStreamsInput ne reçoit pas de nouveau message et que l’intervalle de temps est épuisé, son exécution s’arrête.

Advanced settings

Consumer properties

Ajoutez les propriétés de consommation Kafka nécessaires pour personnaliser cette table.

Pour plus d’informations concernant les propriétés de consommation à définir dans cette table, consultez la documentation de MapR Streams à l’adresse suivante MapR Streams Overview (en anglais).

Timeout precision(ms)

Saisissez, en millisecondes, la durée à la suite de laquelle vous souhaitez retourner une exception de suspension si aucun message n’est disponible à la consommation.

La valeur -1 indique qu’aucune suspension n’est configurée.

Load the offset with the message

Cochez cette case pour écrire en sortie les offsets des messages consommés au composant suivant. Lorsque vous cochez cette case, une colonne offset en lecture seule est ajoutée au schéma.

Custom encoding

Il est possible de rencontrer des problèmes d’encodage lorsque vous traitez les données stockées. Dans ce cas, cochez cette case pour afficher la liste Encoding.

Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement.

Global Variables

Global Variables

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est utilisé en tant que composant de début et nécessite un lien de sortie. Lorsque le topic MapR Streams à utiliser n’existe pas, vous pouvez d’abord créer ce topic à l’aide du composant <strong>tMapRStreamsCreateTopic</strong> ou de votre interface en ligne de commande MapR.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Prérequis</th>
<th>La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le <em>Studio Talend</em>. La liste suivante présente des informations d’exemple relatives à MapR.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Assurez-vous d’avoir installé le client MapR sur la même machine que le <em>Studio Talend</em> et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées <strong>MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native</strong>. Par exemple, pour Windows, la bibliothèque est <strong>lib\MapRClient.dll</strong> dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : <a href="http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr">http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr</a> (en anglais).</td>
</tr>
<tr>
<td></td>
<td>Si vous n’ajoutez pas de librairie, il est possible que vous rencontrez l’erreur suivante : no MapRClient in java.library.path.</td>
</tr>
<tr>
<td></td>
<td>• Configurez l’argument <strong>-Djava.library.path</strong>, par exemple, dans la zone <strong>Job Run VM arguments</strong> de la vue <strong>Run/Debug</strong> de la boîte de dialogue <strong>[Preferences]</strong> dans le menu <strong>Window</strong>. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (<strong>Data viewer</strong>) afin de visualiser localement dans le studio les données stockées dans MapR.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.</td>
</tr>
</tbody>
</table>
Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tMapRStreamsOutput

Ce composant publie des messages dans un système MapR Streams donné. Ce composant supporte uniquement les versions 5.2 et suivantes de MapR.

Ce composant reçoit des messages sérialisés en tableaux (byte arrays) par le composant précédent et envoie ces messages à un système MapR Streams donné.

Propriétés du tMapRStreamsOutput Standard

Ces propriétés sont utilisées pour configurer le tMapRStreamsOutput s’exécutant dans le framework de Jobs Standard.

Le composant tMapRStreamsOutput Standard appartient à la famille Internet.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Notez que le schéma de ce composant est en lecture seule. Il stocke les messages à publier. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et, dans la liste qui s'affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.</td>
</tr>
</tbody>
</table>
| Distribution et Version | Sélectionnez la distribution MapR à utiliser. Seules les versions 5.2 et supérieures de MapR sont supportées par les composants MapRDB.
Si la distribution que vous devez utiliser pour votre base de données MapRDB n’est pas officiellement supportée par ce composant MapRBD, c’est-à-dire, si la distribution de MapR ne s’affiche pas dans la liste déroulante Version de ce composant ou si cette distribution n’est pas MapR, sélectionnez Custom.
1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,
2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différent éléments Hadoop et le fichier d’index de ces bibliothèques.
Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip. |
de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste [Hadoop configuration](en anglais) et utiliser directement dans votre connexion. Cependant, avec l'évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d'utiliser l'option [Import from existing version](en anglais), afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par [Talend](en anglais). [Talend](en anglais) et sa Communauté fournissent l'opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d'Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d'importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez [Connexion à une distribution Hadoop personnalisée](en anglais) à la page 1677.

| **Topic name** | Saisissez le nom du topic dans lequel vous souhaitez publier les messages. Ce topic doit déjà exister. Vous devez saisir le nom du flux auquel appartient ce topic. La syntaxe est la suivante : `chemin_du_flux:nom_du_topic` |
| **Compress the data** | Cochez la case **Compress the data** afin de compresser les données de sortie. |

**Advanced settings**

<p>| <strong>Producer properties</strong> | Ajoutez dans cette table les propriétés producteur MapR Streams Kafka à personnaliser. Pour plus d'informations concernant la configuration du producteur que vous pouvez définir dans cette table, consultez la section décrivant les propriétés importantes de configuration du producteur pour MapR Streams dans la documentation MapR à l'adresse [MapR Streams Overview](en anglais). |</p>
<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour rassembler les métadonnées de traitement au niveau du Job ainsi qu’au niveau de chaque composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global Variables</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Utilisation</strong></td>
<td></td>
</tr>
<tr>
<td>Règle d’utilisation</td>
<td>Ce composant est un composant de fin. Il nécessite un tJavaRow ou un tJava pour transformer les données entrantes en tableaux (byte arrays) sérialisés. L’exemple suivant vous montre comment construire une instruction pour effectuer cette transformation :</td>
</tr>
<tr>
<td></td>
<td>output_row.serializedValue = input_row.users.getBytes();</td>
</tr>
<tr>
<td></td>
<td>Dans ce code, la variable output_row représente le schéma des données à écrire en sortie dans le tMapRStreamsOutput et output_row.serializedValue la colonne unique en lecture seule de ce schéma. La variable input_row représente le schéma des données entrantes et input_row.users la colonne d’entrée nommée users à transformer en tableaux (byte arrays) par la méthode getBytes().</td>
</tr>
<tr>
<td>Prérequis</td>
<td>La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.</td>
</tr>
<tr>
<td></td>
<td>• Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque...</td>
</tr>
</tbody>
</table>
Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante: 

```no
MapRClient in java.library.path.
```

- Configurez l’argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]** dans le menu **Window**. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tMarketoBulkExec**

Ce composant importe des prospects ou des objets personnalisés dans Marketo à partir d’un fichier local, en mode REST API.

**Propriétés du tMarketoBulkExec Standard**

Ces propriétés sont utilisées pour configurer le composant tMarketoBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tMarketoBulkExec Standard appartient à deux familles : Business et Cloud.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Type</strong></td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td><strong>Connection component</strong></td>
<td>Sélectionnez dans la liste déroulante le composant duquel utiliser les informations de connexion pour configurer la connexion à Marketo.</td>
</tr>
<tr>
<td><strong>Endpoint address</strong></td>
<td>Saisissez l’URL de l’endpoint de l’API du Service Web Marketo. Cette URL se trouve dans le panneau <strong>Admin &gt; Web Services</strong> de Marketo.</td>
</tr>
<tr>
<td><strong>Client access ID</strong></td>
<td>Saisissez l’ID client pour accéder au Service Web Marketo.</td>
</tr>
<tr>
<td></td>
<td>L’ID et la phrase secrète du client se trouvent dans le panneau <strong>Admin &gt; LaunchPoint</strong> de Marketo, en cliquant sur <strong>View Details</strong> du service auquel accéder. Vous pouvez également contacter le Support de Marketo via <a href="mailto:support@marketo.com">support@marketo.com</a>, pour plus d’informations.</td>
</tr>
<tr>
<td><strong>Secret key</strong></td>
<td>Saisissez la phrase secrète du client pour accéder au Service Web Marketo Web.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ <strong>Secret key</strong>, puis, dans la boîte de dialogue...</td>
</tr>
</tbody>
</table>
qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.

<table>
<thead>
<tr>
<th>Schema et Edit schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Un schéma est une description de lignes. Il définit le nombre de champs (colonnes) à traiter et à passer au composant suivant. Le schéma de ce composant est prédéfini et est légèrement différent selon le type de données (Leads ou CustomObjects) à importer. Vous pouvez cliquer sur le bouton à côté du champ Edit schema pour voir son schéma.</td>
</tr>
<tr>
<td>• batchId : ID du lot.</td>
</tr>
<tr>
<td>• importId (pour Leads uniquement) : ID de l’import.</td>
</tr>
<tr>
<td>• importTime (pour CustomObjects uniquement) : temps passé sur le lot.</td>
</tr>
<tr>
<td>• message : message du statut du lot.</td>
</tr>
<tr>
<td>• numOfLeadsProcessed/numOfObjectsProcessed : nombre de lignes traitées jusqu’à présent.</td>
</tr>
<tr>
<td>• numOfRowsFailed : nombre de lignes en échec jusqu’à présent.</td>
</tr>
<tr>
<td>• numOfRowsWithWarning : nombre de lignes avec un avertissement jusqu’à présent.</td>
</tr>
<tr>
<td>• objectApiName (pour CustomObjects uniquement) : nom de l’API de l’objet personnalisé.</td>
</tr>
<tr>
<td>• operation (pour CustomObjects uniquement) : type de l’opération de masse.</td>
</tr>
<tr>
<td>• status : statut du lot.</td>
</tr>
<tr>
<td>• failuresLogFile : log des échecs.</td>
</tr>
<tr>
<td>• warningsLogFile : log des avertissements.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Import to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez le type des données à importer, Leads ou CustomObjects.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez le type du fichier contenant les données à importer, csv, tsv ou ssv.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lookup field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez le champ lookup utilisé pour le dédoublonnage. Cette propriété est disponible uniquement lorsque l’option Leads est sélectionnée dans la liste Import to.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>List Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spécifiez l’ID de la liste statique dans laquelle importer. Cette propriété est disponible uniquement lorsque l’option Leads est sélectionnée dans la liste Import to.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Partition name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spécifiez le nom de la partition des prospects dans laquelle importer. Cette propriété est disponible uniquement lorsque l’option Leads est sélectionnée dans la liste Import to.</td>
</tr>
<tr>
<td>CustomObject name</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>Bulk file path</td>
</tr>
<tr>
<td>Poll wait time in seconds</td>
</tr>
</tbody>
</table>
| Batch log download path | Spécifiez le chemin vers le(s) fichier(s) de log de lot à télécharger, y compris le fichier de log des échecs et celui des avertissements.  
Lors de l’import de prospects, le fichier de log des échecs sera nommé bulk_leads_<batchId>_failures.csv et le fichier de log des avertissements sera nommé bulk_leads_<batchId>_warnings.csv.  
Lors de l’import d’objets personnalisés, le fichier de log des échecs sera nommé bulk_customobjects_<customObjectName>_<batchId>_failures.csv et le fichier de log des avertissements sera nommé bulk_customobjects_<customObjectName>_<batchId>_warnings.csv. |
| Die on error      | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.                                                                                                                       |

**Advanced settings**

| API Mode   | Spécifiez le mode de l’API Marketo. Notez que seul le mode REST API est disponible pour ce composant.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeout</td>
<td>Saisissez le délai avant suspension (en millisecondes) de la connexion au Service Web Marketo, avant de terminer la tentative.</td>
</tr>
<tr>
<td>Max reconnection attempts</td>
<td>Saisissez le nombre maximal de tentatives de connexion au Service Web Marketo avant abandon.</td>
</tr>
<tr>
<td>Attempt interval time</td>
<td>Saisissez la période de temps (en millisecondes) entre deux tentatives de connexion subséquentes.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>
Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER_CALL</td>
<td>Nombre d’appels. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en standalone ou en tant que composant de début dans un Job ou un sous-job. |

Scénario associé

Aucun scénario n’est disponible pour ce composant.
tMarketoConnection

Ce composant ouvre une connexion à Marketo qui peut ensuite être réutilisée par d'autres composants Marketo.

Propriétés du tMarketoConnection Standard

Ces propriétés sont utilisées pour configurer le tMarketoConnection s'exécutant dans le framework de Job Standard.

Le composant tMarketoConnection Standard appartient à deux familles : Business et Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

**Endpoint address**


**Client access ID**

Saisissez l’ID client pour accéder au Service Web Marketo.

L’ID et la phrase secrète du client se trouvent dans le panneau Admin > LaunchPoint de Marketo, en cliquant sur View Details du service auquel accéder. Vous pouvez également contacter le Support de Marketo via support@marketo.com, pour plus d’informations.

**Secret key**

Saisissez la phrase secrète du client pour accéder au Service Web Marketo Web.

Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.
Advanced settings

| API Mode | Sélectionnez le mode d’API Marketo, REST API (par défaut) ou SOAP API.  
| Timeout | Saisissez le délai avant suspension (en millisecondes) de la connexion au Service Web Marketo, avant de terminer la tentative. |
| Max reconnection attempts | Saisissez le nombre maximal de tentatives de connexion au Service Web Marketo avant abandon. |
| Attempt interval time | Saisissez la période de temps (en millisecondes) entre deux tentatives de connexion subséquentes. |
| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Global Variables

| NUMBER_CALL | Nombre d’appels. Cette variable est une variable After et retourne un entier. |
| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Marketo. Dans ce cas, il ouvre une connexion qui peut être réutilisée par ces composants. |

Scénario associé

Pour un scénario associé, consultez Transmission de données avec Marketo et à l’aide d’une API REST à la page 2215.
tMarkettoInput

Ce composant récupère des enregistrements de prospects, l'historique des activités, les modifications de prospects et les données liées aux objets personnalisés de Marketo.

Propriétés du tMarkettoInput Standard

Ces propriétés sont utilisées pour configurer le tMarkettoInput s’exécutant dans le framework de Jobs Standard.

Le composant tMarkettoInput Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-In</strong></td>
<td>les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.

<table>
<thead>
<tr>
<th>Connection component</th>
<th>Sélectionnez dans la liste déroulante le composant duquel utiliser les informations de connexion pour configurer la connexion à Marketo.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint address</td>
<td>Saisissez l’URL de l’endpoint de l’API du Service Web Marketo. Cette URL se trouve dans le panneau Admin &gt; Web Services de Marketo.</td>
</tr>
<tr>
<td>Client access ID</td>
<td>Saisissez l’ID client pour accéder au Service Web Marketo. L’ID et la phrase secrète du client se trouvent dans le panneau Admin &gt; LaunchPoint de Marketo, en cliquant sur View Details du service auquel accéder. Vous pouvez également contacter le Support de Marketo via <a href="mailto:support@marketo.com">support@marketo.com</a>, pour plus d’informations.</td>
</tr>
<tr>
<td>Secret key</td>
<td>Saisissez la phrase secrète du client pour accéder au Service Web Marketo Web. Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue</td>
</tr>
</tbody>
</table>
qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.

<table>
<thead>
<tr>
<th>Schema et Edit schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé ligne lors du nommage des champs.</td>
</tr>
<tr>
<td>- <strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Le schéma de ce composant est prédéfini et sera différent selon le mode API sélectionné, l’opération choisie ou l’élément personnalisé spécifié.</td>
</tr>
<tr>
<td>Notez que lorsque le mode API est modifié, vous devez rafraîchir les colonnes du schéma manuellement en sélectionnant des composants différents dans la liste déroulante <strong>Connection Component</strong> (uniquement lorsque vous réutilisez la connexion créée par un composant tMarketoConnection) et en modifiant l’opération à effectuer dans la liste déroulante <strong>Operation</strong>.</td>
</tr>
<tr>
<td>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td>- <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td>- <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
<tr>
<td>- <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <strong>No</strong> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td>Notez que la fonctionnalité de schéma dynamique pour ce composant est supportée uniquement en mode REST API.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez l’opération à effectuer dans la liste déroulante des opérations.</td>
</tr>
<tr>
<td>- <strong>getLead</strong> : Cette option récupère les informations des prospects et de leur activité dans la base de données Marketo.</td>
</tr>
<tr>
<td>- <strong>getMultipleLeads</strong> : Cette opération récupère les enregistrements des prospects dans des lots.</td>
</tr>
</tbody>
</table>
• **getLeadActivity** : Cette opération récupère l'historique des enregistrements d'activité pour un prospect unique identifié par la clé fournie.

• **getLeadChanges** : Cette opération vérifie les changements des données de prospects dans la base de données Marketo.

• **CustomObject** : Sélectionnez cet élément pour effectuer une action spécifique sur un objet personnalisé. Cet élément est disponible uniquement en mode REST API.

**Mappings**

Renseignez cette table pour mapper le nom des colonnes du schéma du composant au nom des colonnes dans Marketo. Par défaut, les noms de colonnes dans les champs Column name sont les mêmes que dans le schéma.

Notez que, comme certains noms de colonnes dans Marketo peuvent contenir des espaces blancs, ce qui n'est pas autorisé dans le schéma du composant, vous devez spécifier les champs des colonnes qui correspondent, dans la colonne Marketo column name. Si les noms définis des colonnes du schéma sont les mêmes que dans Marketo, il n'est pas nécessaire de configurer le mapping des colonnes.

Cette table est indisponible lorsque l'option CustomObject est sélectionnée dans la liste Operation, en mode REST API.

**Lead selector**

Sélectionnez dans la liste l'un des types de sélecteur de prospects :

• **LeadKeySelector** : sélectionnez ce type afin de récupérer les enregistrements des prospects identifiés par la LeadKey spécifiée.

• **StaticListSelector** : sélectionnez ce type afin de récupérer les enregistrements correspondant à des membres de la liste statique spécifiée. Lorsque cette option est sélectionnée, vous devez spécifier la liste statique par son nom ou ID. Pour ce faire, sélectionnez STATIC_LIST_NAME ou STATIC_LIST_ID dans la liste List param et saisissez la valeur de clé du nom ou de l'ID dans le champ List param value.

• **LastUpdateAtSelector** : sélectionnez ce type afin de récupérer les enregistrements des prospects ayant été mis à jour à des dates spécifiques. Cet élément est disponible uniquement en mode SOAP API.

Cette liste est disponible uniquement lorsque l'option getMultipleLeads est sélectionnée dans la liste Operation.

**Lead key type**

Sélectionnez dans la liste un champ par lequel effectuer les requêtes sur le prospect. Cette liste est disponible lorsque :

• **getLead** est sélectionnée dans la liste Operation, ou
### Lead key value
Spécifiez la valeur du champ par lequel effectuer des requêtes sur le prospect. Ce champ est disponible lorsque :
- `getLead` est sélectionné dans la liste **Operation**, ou
- `getLeadActivity` est sélectionné dans la liste **Operation** en mode SOAP API.

### Lead key values
Spécifiez la liste des valeurs des clés. Ce champ est disponible lorsque :
Cette liste s’affiche uniquement lorsque l’option `getMultipleLeads` est sélectionnée dans la liste **Operation** et **Lead Key Selector** dans la liste **Lead Selector**.

### Set Include Types
Cochez cette case et, dans la table **Activity**, spécifiez le type d’activité que vous souhaitez récupérer, en cliquant sur le bouton `[+]` pour ajouter autant de lignes que nécessaire, chaque ligne pour un type d’activité. Cliquez ensuite à droite de chaque cellule et sélectionnez un type d’activité dans la liste déroulante. Cette case est disponible lorsque :
- l’option `getLeadActivity` est sélectionnée dans la liste **Operation**, ou
- l’option `getLeadChanges` est sélectionnée dans la liste **Operation** en mode SOAP API.

### Set Exclude Types
Cochez cette case et, dans la table **Activity**, spécifiez le type d’activité que vous ne souhaitez pas récupérer, en cliquant sur le bouton `[+]` pour ajouter autant de lignes que nécessaire, chaque ligne pour un type d’activité. Cliquez ensuite à droite de chaque cellule et sélectionnez un type d’activité dans la liste déroulante. Cette case est disponible lorsque :
- l’option `getLeadActivity` est sélectionnée dans la liste **Operation**, ou
- l’option `getLeadChanges` est sélectionnée dans la liste **Operation** en mode SOAP API.

### Oldest update date
Saisissez l’heure de la mise à jour la plus ancienne afin de récupérer uniquement les prospects mis à jour depuis le moment spécifié.
Ce champ est disponible uniquement en mode SOAP API, lorsque l’option `getMultipleLeads` est sélectionnée dans la liste **Operation** et que l’option...
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Update At Selector</td>
<td>Saisissez la date de la dernière mise à jour afin de ne récupérer que les prospects mis à jour avant le moment spécifié. Supporte le format de date supporté par Marketo. Pour plus d'informations relatives aux formats de dates supportés, consultez <a href="http://developers.marketo.com/documentation/soap/getmultipleleads/">http://developers.marketo.com/documentation/soap/getmultipleleads/</a> (en anglais). Ce champ est disponible uniquement en mode SOAP API, lorsque l’option getMultipleLeads est sélectionnée dans la liste Operation et que l’option LastUpdateAtSelector est sélectionnée dans la liste Lead selector.</td>
</tr>
<tr>
<td>Oldest Create Date</td>
<td>Saisissez la date et l’heure de la première création afin de récupérer les données depuis le moment spécifié. Le composant supporte le format de date supporté pour Marketo. Ce champ est disponible uniquement en mode SOAP API, lorsque l’option getLeadChanges est sélectionnée dans la liste Operation.</td>
</tr>
<tr>
<td>Latest Creation Date</td>
<td>Saisissez la date et l’heure de la dernière création afin de récupérer les données avant le moment spécifié. Le composant supporte le format de date supporté par Marketo. Les dates Oldest Create Date et Latest Creation Date peuvent être spécifiées ensemble ou séparément. Ce champ est disponible uniquement en mode SOAP API, lorsque l’option getLeadChanges est sélectionnée dans la liste Operation.</td>
</tr>
<tr>
<td>Field List</td>
<td>Saisissez la liste des noms de champs, séparés par une virgule. Ce champ est disponible lorsque l’option getLeadChanges est sélectionnée dans la liste Operation en mode REST API.</td>
</tr>
<tr>
<td>Since DateTime</td>
<td>Saisissez la date à partir de laquelle récupérer les données créées. Ce champ est disponible lorsque l’option getLeadActivity ou getLeadChanges est sélectionnée dans la liste Operation en mode REST API.</td>
</tr>
<tr>
<td>Action</td>
<td>Sélectionnez l’opération à effectuer sur l’objet personnalisé spécifié. • <strong>describe</strong> : retourne les métadonnées d’un objet personnalisé donné. • <strong>list</strong> : retourne une liste de types d’objets personnalisés disponibles dans l’instance cible, avec les informations d’ID et de dédoublonnage pour chaque type. Lorsque cette option est sélectionnée, vous devez spécifier une liste de noms séparés par une virgule.</td>
</tr>
</tbody>
</table>
une virgule afin de filtrer les types dans le champ **CustomObject names**.

- **get** : récupère une liste des enregistrements des objets personnalisés en se basant sur le filtre et un ensemble de valeurs.

Cette liste est disponible uniquement lorsque l’option **CustomObject** est sélectionnée dans la liste **Operation**.

<table>
<thead>
<tr>
<th><strong>CustomObject name</strong></th>
<th>Spécifiez le nom de l’objet personnalisé. Ce champ est disponible uniquement lorsque l’option <strong>CustomObject</strong> est sélectionnée dans la liste <strong>Operation</strong> et que l’option <strong>describe</strong> ou <strong>get</strong> est sélectionnée dans la liste <strong>Action</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fetch schema</strong></td>
<td>Cliquez sur ce bouton pour récupérer le schéma de l’objet personnalisé spécifié. Ce bouton est disponible uniquement lorsque l’option <strong>CustomObject</strong> est sélectionnée dans la liste <strong>Operation</strong> et que l’option <strong>get</strong> est sélectionnée dans la liste <strong>Action</strong>.</td>
</tr>
<tr>
<td><strong>Filter type</strong></td>
<td>Spécifiez le champ sur lequel filtrer. Les champs sur lesquels ils est possible d’effectuer des recherches peuvent être récupérés par une action <strong>describe</strong> avec l’objet personnalisé. Ce champ est disponible uniquement lorsque l’option <strong>CustomObject</strong> est sélectionnée dans la liste <strong>Operation</strong> et que l’option <strong>get</strong> est sélectionnée dans la liste <strong>Action</strong>.</td>
</tr>
<tr>
<td><strong>Filter values</strong></td>
<td>Spécifiez une liste, séparée par une virgule, de valeurs de champs par rapport auxquelles effectuer un rapprochement. Ce champ est disponible uniquement lorsque l’option <strong>CustomObject</strong> est sélectionnée dans la liste <strong>Operation</strong> et que l’option <strong>get</strong> est sélectionnée dans la liste <strong>Action</strong>.</td>
</tr>
</tbody>
</table>
| **Batch Size**       | Limite maximale du nombre des données de prospects à récupérer par lot. Ce champ est disponible si :  
  - l’option **getMultipleLeads** est sélectionnée dans la liste **Operation** et que l’option **StaticListSelector** est sélectionnée dans la liste **Lead Selector**.  
  - l’option **getLeadActivity** ou **getLeadChanges** est sélectionnée dans la liste **Operation**.  
  - L’option **CustomObject** est sélectionnée dans la liste **Operation** et l’option **get** est sélectionnée dans la liste **Action**. |
| **Die on error**     | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. |
### Advanced settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| API Mode                      | Sélectionnez le mode d’API Marketo, REST API (par défaut) ou SOAP API.  
| Timeout                       | Saisissez le délai avant suspension (en millisecondes) de la connexion au Service Web Marketo, avant de terminer la tentative. |
| Max reconnection attempts     | Saisissez le nombre maximal de tentatives de connexion au Service Web Marketo avant abandon.                                                  |
| Attempt interval time         | Saisissez la période de temps (en millisecondes) entre deux tentatives de connexion subséquentes.                                               |
| tStatCatcher Statistics       | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.            |

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER_CALL</td>
<td>Nombre d’appels. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé, consultez Transmission de données avec Marketo et à l’aide d’une API REST à la page 2215.
tMarketoListOperation

Ce composant ajoute ou supprime un ou plusieurs prospect(s) dans une liste dans Marketo. Il vous permet également de vérifier l’existence d’un ou plusieurs prospect(s) dans cette liste.

Propriétés du tMarketoListOperation Standard

Ces propriétés sont utilisées pour configurer le tMarketoListOperation s’exécutant dans le framework de Jobs Standard.
Le composant tMarketoListOperation Standard appartient aux familles Business et Cloud.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste <strong>Connection Component</strong>.</td>
</tr>
</tbody>
</table>

| Connection component | Sélectionnez dans la liste déroulante le composant duquel utiliser les informations de connexion pour configurer la connexion à Marketo. |

| Endpoint address | Saisissez l’URL de l’endpoint de l’API du Service Web Marketo. Cette URL se trouve dans le panneau **Admin > Web Services** de Marketo. |

| Client Access ID | Saisissez l’ID client pour accéder au Service Web Marketo. |
|                 | L’ID et la phrase secrète du client se trouvent dans le panneau **Admin > LaunchPoint** de Marketo, en cliquant sur **View Details** du service auquel accéder. Vous pouvez également contacter le Support de Marketo via **support@marketo.com**, pour plus d’informations. |

| Secret key | Saisissez la phrase secrète du client pour accéder au Service Web Marketo Web. |
|           | Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ **Secret key**, puis, dans la boîte de dialogue |
qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

Notez que la fonctionnalité de schéma dynamique pour ce composant est supportée uniquement en mode REST API.

**Operation**

Sélectionnez dans la liste une opération à effectuer.

- **addTo** : cette option ajoute un ou plusieurs prospect(s) dans une liste.
- **isMemberOf** : cette option recherche dans la base de données si le(s) prospect(s) spécifique(s) existe(nt) dans la liste.
- **removeFrom** : cette opération supprime un ou plusieurs prospect(s) d’une liste.

**Add or remove multiple prospects**

Cochez cette case afin d’ajouter ou de supprimer plusieurs prospect(s) d’une liste dans la base de données Marketo.

Cette case est disponible uniquement si vous avez sélectionné **addTo** ou **removeFrom** dans la liste **Operation**.

**Die on error**

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.

Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.

Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur en utilisant la connexion **Row > Reject** si la case **Add or remove many prospects** est décochée.

**Advanced settings**

| API Mode | Sélectionnez le mode d’API Marketo, **REST API** (par défaut) ou **SOAP API**. |

2206


### Timeout

Saisissez le délai avant suspension (en millisecondes) de la connexion au Service Web Marketo, avant de terminer la tentative.

### Max reconnection attempts

Saisissez le nombre maximal de tentatives de reconnexion au Service Web Marketo avant abandon.

### Attempt interval time

Saisissez la période de temps (en millisecondes) entre deux tentatives de reconnexion subséquentes.

### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER_CALL</td>
<td>Nombre d'appels. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

**Règle d'utilisation**

Ce composant est généralement utilisé comme composant intermédiaire et nécessite un composant d'entrée et un composant de sortie.

### Ajouter un enregistrement de prospect dans une liste Marketo à l'aide de l'API SOAP UI

Le scénario suivant décrit un Job à trois composants qui ajoute un enregistrement de prospect dans une liste spécifiée de Marketo, à l'aide de l'API SOAP UI.
Configurer le Job pour ajouter un enregistrement de prospect dans une liste Marketo

**Procédure**

1. Créez un nouveau Job et ajoutez un composant `tFixedFlowInput`, un `tMarketoListOperation` et un `tLogRow`, en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.
2. Double-cliquez sur le `tMarketoListOperation` pour ouvrir sa vue `Component`, cliquez sur l’onglet `Advanced settings`, pour ouvrir la vue de configuration avancée, puis sélectionnez `SOAP` dans la liste `API Mode`.
3. Reliez le `tFixedFlowInput` au `tMarketoListOperation` à l’aide d’un lien `Row > Main`. Dans la boîte de dialogue qui s’ouvre, cliquez sur `Yes` pour récupérer le schéma du composant cible.
4. Reliez le `tMarketoListOperation` au `tLogRow` à l’aide d’un lien `Row > Main`.

Configurer le Job pour ajouter un enregistrement de prospect dans une liste Marketo

**Procédure**

1. Double-cliquez sur le composant `tFixedFlowInput` pour ouvrir sa vue `Basic settings`.
2. Cliquez sur le bouton ` [...] ` à côté du champ `Edit schema` pour voir son schéma.
3. Dans la zone Mode, sélectionnez Use Inline Table, puis cliquez sur le bouton [+] pour ajouter une ligne. Renseignez cette ligne avec ses valeurs respectives. Dans cet exemple, ces valeurs sont : MKTOLISTNAME pour ListKeyType, Test_Comp_QA pour ListKeyValue, IDNUM pour LeadKeyType et 9300203 pour LeadKeyValue.

4. Double-cliquez sur le tMarkettoListOperation pour ouvrir sa vue Basic settings.

5. Dans le champ Endpoint address, saisissez l’URL de l’endpoint de l’API SOAP Marketo, que vous pouvez trouver dans le panneau Marketo Admin > Web Services.


7. Dans la liste Operation, sélectionnez addTo pour ajouter le prospect d’ID 9300203 dans la liste Test_Comp_QA.

8. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings.

9. Cliquez sur le bouton Sync columns afin de synchroniser le schéma et le schéma d’entrée.

10. Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour une lisibilité optimale des résultats.

Exécuter le Job d’ajout d’un enregistrement de prospect dans une liste Marketo

Procédure

1. Appuyez sur les touches Ctrl + S afin de sauvegarder le Job.

2. Appuyez sur F6 ou cliquez sur le bouton Run de la vue Run pour exécuter le Job.

```plaintext
[statistics] connecting to socket on port 3520
[statistics] connected

-------------------------------
|                              |
| tLogRow_1                    |
|-------------------------------
| ListKeyType | ListKeyValue | LeadKeyType | LeadKeyValue | Success |
| MKTOLISTNAME | Test_Comp_QA | IDNUM | 9300203 | true |

[statistics] disconnected
```

Comme affiché ci-dessus, le résultat de l’opération est affiché dans la console de la vue Run.

3. Vérifiez le résultat dans Marketo.
Le prospect d'ID 9300203 dans Marketo est ajouté à la liste Test_Comp_QA.
**tMarketoOutput**

Ce composant écrit des enregistrements de prospects ou des enregistrements d’objets personnalisés à partir du flux de données entrant dans Marketo.

**Propriétés du tMarketoOutput Standard**

Ces propriétés sont utilisées pour configurer le tMarketoOutput s’exécutant dans le framework de Jobs Standard.

Le composant tMarketoOutput Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste **Connection Component**.

<table>
<thead>
<tr>
<th>Connection component</th>
<th>Sélectionnez dans la liste déroulante le composant duquel utiliser les informations de connexion pour configurer la connexion à Marketo.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint address</td>
<td>Saisissez l’URL de l’endpoint de l’API du Service Web Marketo. Cette URL se trouve dans le panneau <strong>Admin &gt; Web Services</strong> de Marketo.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Client access ID</th>
<th>Saisissez l’ID client pour accéder au Service Web Marketo.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L’ID et la phrase secrète du client se trouvent dans le panneau <strong>Admin &gt; LaunchPoint</strong> de Marketo, en cliquant sur <strong>View Details</strong> du service auquel accéder. Vous pouvez également contacter le Support de Marketo via <a href="mailto:support@marketo.com">support@marketo.com</a>, pour plus d’informations.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Secret key</th>
<th>Saisissez la phrase secrète du client pour accéder au Service Web Marketo Web.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ <strong>Secret key</strong>, puis, dans la boîte de dialogue...</td>
</tr>
</tbody>
</table>
qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le [Guide utilisateur du Studio Talend](#).

- **Repository** : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le [Guide utilisateur du Studio Talend](#).

Le schéma de ce composant est prédéfini et sera différent selon le mode API sélectionné, l’opération choisie ou l’élément personnalisé spécifié.

Notez que lorsque le mode API est modifié, vous devez rafraîchir les colonnes du schéma manuellement en sélectionnant des composants différents dans la liste déroulante `Connection Component` (uniquement lorsque vous réutilisez la connexion créée par un composant `tMarketoConnection`) et en modifiant l’opération à effectuer dans la liste déroulante `Operation`.

Cliquez sur `Edit schema` pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton `Sync columns` pour récupérer le schéma du composant précédent.

Notez que la fonctionnalité de schéma dynamique pour ce composant est supportée uniquement en mode REST API.

### Operation

 Sélectionnez dans la liste une opération vous permettant de synchroniser les données entre Marketo et un système externe.

L’opération est supportée en mode **REST API** et en mode **SOAP API**.
- **syncLead** : cette option sollicite une opération d’insertion ou de mise à jour pour un enregistrement de prospect.
- **syncMultipleLeads** : cette option sollicite une opération d’insertion ou de mise à jour pour plusieurs enregistrements de prospects regroupés par lots.
- **deleteLeads** (pour REST uniquement) : supprime une liste de prospects.
- **syncCustomObjects** (pour REST uniquement) : insère, met à jour ou insère et met à jour des enregistrements d’objets personnalisés dans l’instance cible.
- **deleteCustomObjects** (pour REST uniquement) : supprime un ensemble d’enregistrements d’objets personnalisés.

### Operation type

 Sélectionnez dans la liste un type d’opération de synchronisation à effectuer.

- **createOnly** : crée de nouveaux prospects.
- **updateOnly** : met à jour des prospects existants.
- **createOrUpdate** : crée de nouveaux prospects ou met à jour les prospects s’ils existent déjà.
- **createDuplicate** : crée de nouveaux prospects. S’ils existent déjà, l’option crée des prospects en doublon.

Cette liste est disponible uniquement en mode REST API lorsque l’option **syncLead** ou **syncMultipleLeads** est sélectionnée dans la liste **Operation**.

### Lookup field

 Sélectionnez le champ lookup utilisé pour trouver le doublon.

Cette liste est disponible uniquement en mode REST API lorsque l’option **syncLead** ou **syncMultipleLeads** est sélectionnée dans la liste **Operation**.

### Mappings

Renseignez cette table pour mapper le nom des colonnes du schéma du composant au nom des colonnes dans Marketo. Par défaut, les noms de colonnes dans les champs **Column name** sont les mêmes que dans le schéma.

Notez que, comme certains noms de colonnes dans Marketo peuvent contenir des espaces blancs, ce qui n’est pas autorisé dans le schéma du composant, vous devez spécifier les champs des colonnes qui correspondent, dans la colonne **Marketo column name**. Si les noms définis des colonnes du schéma sont les mêmes que dans Marketo, il n’est pas nécessaire de configurer le mapping des colonnes.

Cette table est disponible uniquement en mode SOAP API.

### Dedupe prospects record using the email address

Cochez cette case afin de dédoublonner les enregistrements des prospects en utilisant l’adresse e-mail.
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete leads in batch</strong></td>
<td>Cochez cette case pour supprimer les prospects par lots.</td>
</tr>
<tr>
<td></td>
<td>Cette case est disponible uniquement lorsque l’option deleteLeads est sélectionnée dans la liste <strong>Operation</strong>.</td>
</tr>
<tr>
<td><strong>Batch Size</strong></td>
<td>Spécifiez le nombre d’enregistrements à traiter dans chaque lot.</td>
</tr>
<tr>
<td></td>
<td>Ce champ est disponible uniquement si :</td>
</tr>
<tr>
<td></td>
<td>• l’option syncMultipleLeads est sélectionnée dans la liste <strong>Operation</strong>, ou</td>
</tr>
<tr>
<td></td>
<td>• si l’option deleteLeads est sélectionnée dans la liste <strong>Operation</strong> et que la case <strong>Delete leads in batch</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Sync method</strong></td>
<td>Sélectionnez dans la liste un type d’opération de synchronisation à effectuer.</td>
</tr>
<tr>
<td></td>
<td>Cette liste est disponible uniquement lorsque l’option syncCustomObjects est sélectionnée dans la liste <strong>Operation</strong>.</td>
</tr>
<tr>
<td><strong>Dedupe by</strong></td>
<td>Spécifiez le champ sur lequel les enregistrements seront dédoublonnés.</td>
</tr>
<tr>
<td></td>
<td>Ce champ est disponible uniquement lorsque l’option syncCustomObjects est sélectionnée dans la liste <strong>Operation</strong>.</td>
</tr>
<tr>
<td><strong>Delete by</strong></td>
<td>Sélectionnez le champ sur lequel l’opération de suppression sera effectuée.</td>
</tr>
<tr>
<td></td>
<td>Cette liste est disponible uniquement lorsque l’option deleteCustomObjects est sélectionnée dans la liste <strong>Operation</strong>.</td>
</tr>
<tr>
<td><strong>Custom object name</strong></td>
<td>Spécifiez le nom de l’objet personnalisé.</td>
</tr>
<tr>
<td></td>
<td>Ce champ est disponible uniquement lorsque l’option syncCustomObjects ou deleteCustomObjects est sélectionnée dans la liste <strong>Operation</strong>.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.</td>
</tr>
<tr>
<td></td>
<td>Décocochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.</td>
</tr>
<tr>
<td></td>
<td>Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur en utilisant la connexion Row &gt; <strong>Reject</strong>. Toutefois, notez qu’aucune erreur ne peut être récupérée lors de la synchronisation et de la suppression des prospects en lot.</td>
</tr>
</tbody>
</table>
## Advanced settings

| **API Mode** | Sélectionnez le mode d’API Marketo, REST API (par défaut) ou SOAP API.
|
| **Timeout** | Saisissez le délai avant suspension (en millisecondes) de la connexion au Service Web Marketo, avant de terminer la tentative.
|
| **Max reconnection attempts** | Saisissez le nombre maximal de tentatives de reconnexion au Service Web Marketo avant abandon.
|
| **Attempt interval time** | Saisissez la période de temps (en millisecondes) entre deux tentatives de reconnexion subséquentes.
|
| **tStatCatcher Statistics** | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.
|
## Global Variables

| **NUMBER_CALL** | Nombre d’appels. Cette variable est une variable *After* et retourne un entier.
|
| **ERROR_MESSAGE** | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères.
|
## Utilisation

| **Règle d’utilisation** | Ce composant est généralement utilisé comme composant de fin dans un Job ou un sous-job et nécessite un lien d’entrée.
|
## Transmission de données avec Marketo et à l’aide d’une API REST

Voici un exemple d’utilisation de composants Talend permettant de se connecter à Marketo, d’insérer plusieurs enregistrements de prospects dans Marketo, de récupérer ces enregistrements et de les afficher dans la console.
Configurer le Job de transmission de données avec Marketo à l'aide d'une API REST

Procédure

1. Ajoutez un composant `tMarketoConnection`, deux `tFixedFlowInput`, un `tMarketoOutput`, un `tMarketoInput` et un `tLogRow` en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.

2. Assurez-vous d’avoir sélectionné `REST` dans la liste `API Mode` de la vue `Advanced settings` des composants `tMarketoOutput` et `tMarketoInput`.

3. Reliez le premier `tFixedFlowInput` au `tMarketoOutput` à l’aide d’un lien `Row > Main`. Dans la boîte de dialogue qui s’ouvre, cliquez sur `Yes` afin que le premier `tFixedFlowInput` récupère le schéma du `tMarketoOutput` cible.

4. Reliez le second `tFixedFlowInput` au `tMarketoInput` à l’aide d’un lien `Row > Main`. Dans la boîte de dialogue, cliquez sur `No`, pour que le second `tFixedFlowInput` ne récupère pas le schéma du composant `tMarketoInput` cible.

5. Reliez le `tMarketoInput` au `tLogRow` à l’aide d’un lien `Row > Main`.


7. Répétez l’opération pour relier le premier `tFixedFlowInput` au second `tFixedFlowInput`.

Se connecter à Marketo à l’aide d’une API REST

Avant de commencer

Avant de vous connecter à Marketo à l’aide de l’endpoint de l’API REST, vous devez vous assurer qu’un service personnalisé, fournissant les identifiants d’authentification à Marketo et vous permettant d’accéder aux données dans Marketo a bien été créé et associé à un utilisateur avec une API unique par votre administrateur Marketo. Pour plus d’informations concernant les services personnalisés de Marketo (Marketo Custom Service), consultez Custom Services (en anglais).

Procédure

1. Double-cliquez sur le `tMarketoConnection` pour ouvrir sa vue `Basic settings`. 
2. Dans le champ **Endpoint address**, saisissez l’URL de l’endpoint de l’API REST, que vous pouvez trouver dans le panneau **Admin > Web Services** de Marketo.

3. Dans les champs **Client Access ID** et **Secret key**, saisissez l’ID et la phrase secrète du client, requis pour accéder à Marketo.

L’ID et la phrase secrète du client se trouvent dans le panneau **Admin > LaunchPoint** de Marketo, en cliquant sur **View Details** du service auquel accéder.

**Insérer des enregistrements de prospects dans Marketo à l’aide d’une API REST**

**Procédure**

1. Double-cliquez sur le premier **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur du schéma.
3. Dans l’éditeur du schéma, sélectionnez les colonnes id et Status ne nécessitant pas de valeur d’entrée, puis cliquez sur le bouton [x] pour les supprimer. Cliquez sur OK pour sauvegarder ces modifications et fermer l’éditeur.

4. Dans la zone Mode, sélectionnez Use Inline Content(delimited file) et saisissez les données suivantes dans le champ Content.

```
dev@talend.com;Peter;Wang
doc@talend.com;Judy;Zhang
dev@talend.com;Nick;Liu
dev@talend.com;Mark;Zhao
doc@talend.com;Alex;Sun
```

5. Double-cliquez sur le composant tMarketoOutput pour ouvrir sa vue Basic settings.

6. Sélectionnez dans la liste déroulante Connection Component le composant duquel réutiliser les informations de connexion pour configurer la connexion à Marketo, tMarketoConnection_1 dans cet exemple.

7. Dans la liste Operation, sélectionnez SyncMultipleLeads pour insérer plusieurs enregistrements de prospects.

8. Dans la liste Operation type, sélectionnez createDuplicate pour permettre la génération de doublons de prospects.
Récupérer les enregistrements des prospects depuis Marketo à l'aide d'une API REST

Procédure

1. Double-cliquez sur le second tFixedFlowInput pour ouvrir sa vue Basic settings.

2. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans l'éditeur du schéma qui s'affiche, définissez le schéma en ajoutant une colonne mail de type String. Cliquez sur OK pour sauvegarder les modifications et fermer l'éditeur.

3. Dans la zone Mode, sélectionnez Use Inline Table. Cliquez deux fois sur le bouton [+ ] sous la table pour ajouter deux lignes et renseignez ces lignes en saisissant leurs valeurs respectives. Dans cet exemple, deux valeurs d'adresses e-mail dev@talend.com et doc@talend.com sont ajoutées afin que les enregistrements des prospects identifiés par ces valeurs puissent être récupérés.

5. Sélectionnez dans la liste déroulante **Connection Component** le composant duquel utiliser les informations de connexion pour configurer la connexion à Marketo, `tMarketoConnection_1` dans cet exemple.

6. Sélectionnez `getMultipleLeads` dans la liste **Operation** afin de récupérer un lot d’enregistrements de prospects.

7. Sélectionnez `LeadKeySelector` dans la liste **Lead Selector** et sélectionnez `email` dans la liste **LeadKey type**. Sélectionnez `email` dans la liste **LeadKey values** pour récupérer les prospects identifiés par les valeurs d’e-mail passées depuis le second `tFixedFlowInput`.

8. Double-cliquez sur le `tLogRow` pour ouvrir sa vue **Basic settings**.

9. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour une lisibilité optimale des résultats.

---

**Exécuter le Job de transmission de données avec Marketo via une API REST**

**Procédure**

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder le Job.

2. Appuyez sur **F6** pour exécuter le Job.

```plaintext
| [statistics] connecting to socket on port 3375
| [statistics] connected
| tLogRow_1
| [id | email | firstName | lastName | createdAt | updatedAt |
| 10873609 | doc@talend.com | Judy | Zhang | 2016-03-11T02:04:17+0800 | 2016-03-11T02:11:25+0800 |
| 10873650 | dev@talend.com | Peter | Wang | 2016-03-11T02:04:17+0800 | 2016-03-11T02:11:25+0800 |
| 10873600 | dev@talend.com | Nick | Liu | 2016-03-11T02:04:17+0800 | 2016-03-11T02:11:39+0800 |
| 10873607 | dev@talend.com | Mark | Zhao | 2016-03-11T02:04:17+0800 | 2016-03-11T02:11:41+0800 |
| 10873659 | dev@talend.com | Alex | Sun | 2016-03-11T02:04:17+0800 | 2016-03-11T02:11:44+0800 |
| [statistics] disconnected
```

Comme affiché ci-dessus, les enregistrements de prospects insérés dans Marketo sont récupérés et affichés dans la console.
tMarkLogicBulkLoad

Ce composant importe des fichiers locaux sur un serveur d’une base de données MarkLogic en mode bulk, à l’aide de l’outil MarkLogic Content Pump (MLCP).

Propriétés du tMarkLogicBulkLoad Standard

Ces propriétés sont utilisées pour configurer le tMarkLogicBulkLoad s’exécutant dans le framework de Jobs Standard.

Le composant tMarkLogicBulkLoad Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Built-In : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>• Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

Host

Saisissez l’adresse IP ou le nom d’hôte du serveur MarkLogic.

Port

Saisissez le numéro du port d’écoute du serveur MarkLogic.
### Database
Saisissez le nom de la base de données MarkLogic que vous souhaitez utiliser.

### Username et Password
Saisissez les informations d’authentification de l’utilisateur pour accéder à la base de données MarkLogic.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

### Load Folder
Spécifiez le répertoire des fichiers locaux à importer dans la base de données MarkLogic.

### DocId Prefixed As
Spécifiez le préfixe utilisé pour construire l’URI des fichiers importés. Pour plus d’informations concernant la construction d’URI, consultez https://docs.marklogic.com/guide/ingestion/content-pump#id_14982 (en anglais).

### Advanced settings

#### Additional MLCP Parameters
Spécifiez les options supplémentaires pour la commande d’import MLCP. Pour plus d’informations concernant les options d’import, consultez https://docs.marklogic.com/guide/ingestion/content-pump#id_63999 (en anglais).

#### Use external MLCP process
Comme le programme MarkLogic Content Pump (MLCP) est requis sur la machine où vous exécutez le Job utilisant ce composant, vous pouvez :

- laissez décochée la case pour utiliser une bibliothèque MLCP built-in lors de l’exécution de ce composant pour charger des données. Dans ce cas, votre Job demande beaucoup de mémoire.

- si vous devez utiliser votre propre bibliothèque MLCP, c’est-à-dire une bibliothèque MLCP externe au Studio, cochez cette case. Dans ce cas, votre Job lui-même ne demande pas trop de mémoire mais le processus MLCP en demande beaucoup.

Pour plus d’informations concernant MLCP, consultez la documentation Marklogic associée : Loading Content Using Marklogic Content Pump (en anglais).

#### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau des composants.

### Variables globales

#### Global Variables
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé avec le tMarkLogicInput afin de vérifier si les fichiers ont bien été importés, comme attendu.</th>
</tr>
</thead>
</table>

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.  


| Limitation | L’outil MarkLogic Content Pump (MLCP) doit être installé sur la machine sur laquelle vous exécutez le Job impliquant ce composant. |

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tMarkLogicClose

Ce composant ferme une connexion active à une base de données MarkLogic afin de libérer des ressources occupées.

Propriétés du tMarkLogicClose Standard

Ces propriétés sont utilisées pour configurer le tMarkLogicClose s’exécutant dans le framework de Jobs Standard.

Le composant tMarkLogicClose Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Component List | Sélectionnez dans la liste le composant tMarkLogicConnection ouvrant la connexion à fermer. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau des composants. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants MarkLogic, notamment le tMarkLogicConnection. |

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de |
contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tMarkLogicConnection

Ce composant ouvre une connexion à une base de données MarkLogic pouvant être réutilisée par d’autres composants MarkLogic.

Propriétés du tMarkLogicConnection Standard

Ces propriétés sont utilisées pour configurer le tMarkLogicConnection s’exécutant dans le framework de Jobs Standard.

Le composant tMarkLogicConnection Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Property Type          | Peut être Built-in ou Repository.  
|                       | **Built-in** : Propriétés utilisées ponctuellement.  
|                       | **Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.  
| Host                   | Saisissez l’adresse IP ou le nom d’hôte du serveur MarkLogic.  
| Port                   | Saisissez le numéro du port d’écoute du serveur MarkLogic.  
| Database               | Saisissez le nom de la base de données MarkLogic que vous souhaitez utiliser.  
| Username et Password   | Saisissez les informations d’authentification de l’utilisateur pour accéder à la base de données MarkLogic.  
|                        | Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.  
| Authentication Type    | Sélectionnez dans la liste un type d’authentification, **DIGEST** ou **BASIC**. Pour plus d’informations concernant les types d’authentification, consultez **http://docs.marklogic.com/guide/security/authentication#id_56266** (en anglais).  

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau des composants.  

Global Variables

Global Variables

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant est généralement utilisé avec d’autres composants MarkLogic.

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
Ce composant recherche le contenu d'un document dans une base de données MarkLogic en se basant sur une requête.

Le tMarkLogicInput passe l'URI et le contenu du document au composant suivant via un lien Row > Main.

**Propriétés du tMarkLogicInput Standard**

Ces propriétés sont utilisées pour configurer le tMarkLogicInput s’exécutant dans le framework de Jobs Standard.

Le composant tMarkLogicInput Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Basic settings**

| Property Type | Peut être **Built-in** ou **Repository**.  
**Built-in** : Propriétés utilisées ponctuellement.  
**Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées. |
|----------------|------------------------------------------------------|
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.  
**Remarque** :  
Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :  
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.  
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.  
Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend. |
<p>| Host | Saisissez l'adresse IP ou le nom d'hôte du serveur MarkLogic. |</p>
<table>
<thead>
<tr>
<th>Port</th>
<th>Saisissez le numéro du port d'écoute du serveur MarkLogic.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données MarkLogic que vous souhaitez utiliser.</td>
</tr>
</tbody>
</table>
| Username et Password | Saisissez les informations d'authentification de l'utilisateur pour accéder à la base de données MarkLogic.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Authentication | Sélectionnez dans la liste un type d'authentification, DIGEST ou BASIC. Pour plus d'informations concernant les types d'authentification, consultez http://docs.marklogic.com/guide/security/authentication#id_56266 (en anglais). |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton [...] à côté du champ Edit schema pour voir le schéma prédéfini contenant les deux colonnes suivantes :  
• docId : URI du document.  
• docContent : contenu du document. |
| Query Criteria | Spécifiez la requête sur laquelle baser la recherche. Pour plus d'informations concernant la requête, consultez https://docs.marklogic.com/guide/search-dev/string-query (en anglais). |

**Advanced settings**

<table>
<thead>
<tr>
<th>Maximal Retrieve Number</th>
<th>Saisissez le nombre maximal de documents à récupérer.</th>
</tr>
</thead>
</table>
| Use Query Option | Cochez cette case afin de personnaliser la requête avec une option de requête. Cette option sera stockée sur le serveur REST et pourra être référencée par son nom par les autres requêtes. Pour plus d'informations concernant les options de requête, consultez http://docs.marklogic.com/guide/java/query-options#id_20346 (en anglais).  
**Query Literal Type** : sélectionnez un type de présentation duquel créer l'option de requête, XML ou JSON.  
**Query Option Name** : spécifiez le nom de l'option de requête. |
<table>
<thead>
<tr>
<th>Query Option Literals</th>
<th>implémentez l’option de requête pour une représentation depuis un XML ou JSON brut, selon le type de représentation sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau des composants.</td>
</tr>
</tbody>
</table>

### Global Variables

| **Global Variables** | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

### Utilisation

<table>
<thead>
<tr>
<th><strong>Règle d’utilisation</strong></th>
<th>Ce composant est généralement utilisé en tant que composant de début et nécessite un flux de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <em>Dynamic settings</em> est disponible uniquement lorsque la case <em>Use an existing connection</em> est cochée dans la vue <em>Basic settings</em>. Lorsqu’un paramètre dynamique est configuré, la liste <em>Component List</em> de la vue <em>Basic settings</em> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les</td>
</tr>
</tbody>
</table>
Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tMarkLogicOutput

Ce composant crée, met à jour ou supprime des contenus de documents dans une base de données MarkLogic.

Le tMarkLogicOutput vous permet de maintenir des contenus de documents dans une base de données MarkLogic selon les opérations prédéfinies et le flux d’entrée provenant du composant précédent.

Propriétés du tMarkLogicOutput Standard

Ces propriétés sont utilisées pour configurer le tMarkLogicOutput s’exécutant dans le framework de Jobs Standard.

Le composant tMarkLogicOutput Standard appartient à la famille Big Data.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Property Type | Peut être Built-in ou Repository.  
**Built-in** : Propriétés utilisées ponctuellement.  
**Repository** : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |
|---------------|--------------------------------------------------------------------------------|
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
**Remarque** :  
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :  
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.  
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.  
Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend. |
<p>| Host | Saisissez l’adresse IP ou le nom d’hôte du serveur MarkLogic. |</p>
<table>
<thead>
<tr>
<th>Port</th>
<th>Saisissez le numéro du port d’écoute du serveur MarkLogic.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données MarkLogic que vous souhaitez utiliser.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d’authentification de l’utilisateur pour accéder à la base de données MarkLogic. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Authentication</td>
<td>Sélectionnez dans la liste un type d’authentification, DIGEST ou BASIC. Pour plus d’informations concernant les types d’authentification, consultez <a href="http://docs.marklogic.com/guide/security/authentication#id_56266">http://docs.marklogic.com/guide/security/authentication#id_56266</a> (en anglais).</td>
</tr>
</tbody>
</table>
| Action | Sélectionnez l’opération à effectuer  
• UPSERT : créer des documents s’ils n’existent pas ou mettre à jour le contenu de documents existants.  
• PATCH : effectuer une mise à jour partielle du contenu des documents.  
• DELETE : supprimer les documents correspondants au flux d’entrée.  
Notez que lorsque l’opération DELETE est sélectionnée dans la liste Action, le schéma d’entrée doit contenir au moins une colonne docId décrivant l’URI des documents à supprimer. Les autres colonnes, s’il y en a, seront ignorées. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
Le schéma de ce composant est en lecture seule. Vous pouvez cliquer sur le bouton [...] à côté du champ Edit schema pour voir le schéma prédéfini contenant les deux colonnes suivantes :  
• docId : URI du document.  
• docContent : contenu du document. |
| Advanced settings |  |
| Doc Type | Sélectionnez le type des documents à traiter : MIXED, PLAIN TEXT, JSON, XML ou BINARY. |
| Auto Generate Doc ID | Cochez cette case pour générer automatiquement les URI des documents et, dans le champ Doc Id Prefix qui |
s’affiche, saisissez le préfixe utilisé pour construire les URI des documents.

Cette case est disponible uniquement lorsque l’opération **UPsert** est sélectionnée dans la liste **Action** et que le type **MIXED** n’est pas sélectionné dans la liste **Doc Type**.

- Si cette case est cochée, le schéma d’entrée doit contenir une colonne **docContent** décrivant le contenu du document et les autres colonnes seront ignorées, s’il y en a.
- Si cette case est décochée, le schéma d’entrée doit contenir deux colonnes, **docId** et **docContent**, décrivant l’URI et le contenu du document.

### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau des composants.

### Global Variables

#### Global Variables

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

#### Règle d’utilisation

Ce composant est généralement utilisé comme composant de fin et nécessite un flux d’entrée.

#### Dynamic settings

Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée.
Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tMaxDBInput

Ce composant lit une table de base de données et en extrait des champs à l’aide de requêtes.

Le tMaxDBInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

**Propriétés du tMaxDBInput Standard**

Ces propriétés sont utilisées pour configurer le tMaxDBInput s’exécutant dans le framework de Jobs Standard.

Le composant tMaxDBInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.

Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host name</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>

| Username et Password | Informations d’authentification de l’utilisateur de base de données. |
|----------------------|-----------------------------------------------------------------
|                      | Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, |
évitez le mot réservé `line` lors du nommage des champs.

**Built-in** : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.


Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Table name**
Saisissez le nom de la table.

**Query type et Query**
Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

**Guess Query**
Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.

**Guess schema**
Cliquez sur le bouton pour récupérer le schéma de la table.

**Advanced settings**

**Trim all the String/Char columns**
Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.

**Trim column**
Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.
**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. |
| Limitation | Ce composant requiert l’installation des fichiers .jar liés. |

**Scénarios associés**

Pour un scénario associé, consultez les scénarios dans :
tMaxDBOutput

Ce composant écrit, met à jour, modifie ou supprime les données d’une base de données.
Le tMaxDBOutput exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Propriétés du tMaxDBOutput Standard

Ces propriétés sont utilisées pour configurer le tMaxDBOutput s’exécutant dans le framework de Jobs Standard.
Le composant tMaxDBOutput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Host
Adresse IP du serveur de base de données.

Port
Numéro du port d’écoute du serveur.

Database
Nom de la base de données.

Username et Password
Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

Table
Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.

Action on table
Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :
None : n'effectuer aucune opération de table.

Drop and create the table : supprimer la table puis en créer une nouvelle.

Create a table : créer une table qui n'existe pas encore.

Create table if doesn't exist : créer la table si nécessaire.

Drop a table if exists and create : supprimer la table si elle existe déjà, puis en créer une nouvelle.

Clear a table : supprimer le contenu de la table.

Truncate table : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.

### Action on data

Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

- **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.
- **Update** : Mettre à jour les entrées existantes.
- **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
- **Delete** : Supprimer les entrées correspondantes au flux d’entrée.

⚠️ **Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations Update et Delete. Pour cela, cliquez sur le bouton [...] à côté du champ Edit Schema et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet Advanced settings pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case Use field options et sélectionnez la case Key in update correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases Key in delete pour les opérations de suppression (Delete).

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects.

**Advanced settings**

**Commit every**

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

**Additional Columns**

Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.
**Name** : Saisissez le nom de la colonne à modifier ou à insérer.

**SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.

**Position** : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l’action à effectuer sur la colonne de référence.

**Reference column** : Saisissez une colonne de référence que le composant tMaxDBOutput peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

**Use field options** : Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.

**Enable debug mode** : Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.

**tStatCatcher Statistics** : Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données MaxDB. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput. |
| Limitation | Ce composant requiert l’installation des fichiers .jar liés. |

**Scénarios associés**

Pour un scénario associé, consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667 du composant tMysqlOutput.
tMaxDBRow

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler), selon la nature de la requête et de la base de données.

tMaxDBRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Propriétés du tMaxDBRow Standard

Ces propriétés sont utilisées pour configurer le tMaxDBRow s'exécutant dans le framework de Jobs Standard.

Le composant tMaxDBRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réserve line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th>Table name</th>
<th>Saisissez le nom de la table.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query type</td>
<td>La requête peut être <strong>Built-in</strong> ou distante dans le Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez la requête appropriée dans le Repository. Le champ <strong>Query</strong> est renseigné automatiquement.</td>
</tr>
<tr>
<td>Query</td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td><strong>Guess Query</strong></td>
<td>Cliquez sur le bouton <strong>Guess Query</strong> pour générer la requête correspondant au schéma de votre table dans le champ <strong>Query</strong>.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien <strong>Row &gt; Rejects</strong>.</td>
</tr>
</tbody>
</table>

### Advanced settings

- **Propagate QUERY’s recordset** : Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.
- **Use PreparedStatement** : Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de
tMaxDBRow

<table>
<thead>
<tr>
<th>Commit every</th>
<th>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL. |
| Limitation | Ce composant requiert l’installation des fichiers .jar liés. |
Scénario associé

Aucun scénario n'est disponible pour ce composant.
tMDMBulkLoad

Ce composant utilise le mode bulk pour écrire en masse des données maître structurées XML dans le.
Notez que le composant tMDMBulkLoad ne génère pas d’entrée de journal en tant que partie du chargement de masse.

Propriétés du tMDMBulkLoad Standard

Ces propriétés sont utilisées pour configurer le tMDMBulkLoad s’exécutant dans le framework de Jobs Standard.
Le composant tMDMBulkLoad Standard appartient à la famille Talend MDM.
Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|                       | • View schema : sélectionnez cette option afin de voir le schéma.
|                       | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
|                       | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. Cliquez sur le bouton Sync columns afin de récupérer le schéma du composant précédent.


XML Field | Sélectionnez le nom de la colonne dans laquelle vous souhaitez écrire les données XML.
<table>
<thead>
<tr>
<th><strong>URL</strong></th>
<th>Saisissez l’URL d’accès au serveur MDM.</th>
</tr>
</thead>
</table>
| **Username et Password** | Informations d’authentification de l’utilisateur du serveur MDM.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Data Model**     | Saisissez le nom du modèle de données par rapport auquel vous souhaitez valider les données à écrire. |
| **Data Container** | Saisissez le nom du conteneur de données contenant les données que vous souhaitez écrire. |
| **Entity**         | Saisissez le nom de l’entité contenant les données que vous souhaitez effacer. |
| **Type**           | Sélectionnez Master ou Staging pour spécifier sur quelle base de données effectuer l’action. |
| **Validate**       | Cochez cette case afin de valider les données que vous souhaitez écrire sur le serveur MDM, par rapport à des règles de validation définies pour le modèle de données courant.  
Pour le conteneur de données PROVISIONING, les vérifications de validation sont toujours effectuées sur les enregistrements entrants, que la case soit cochée ou non.  
Pour plus d’informations concernant la définition des règles de validation, consultez le Guide utilisateur du Studio Talend.  
**Avertissement :**  
Si vous préférez un chargement plus rapide, ne cochez pas cette case. |
| **Generate ID**      | Cochez cette case pour générer un identifiant aux données écrites.  
**Avertissement :**  
Si vous préférez un chargement plus rapide, ne cochez pas cette case. |
| **Insert only**     | Cochez cette case pour passer l’étape qui vérifie si les enregistrements de données à insérer existent déjà sur le serveur MDM, augmentant ainsi les performances.  
Néanmoins, avant d’utiliser cette option, vous devez vous assurer que les enregistrements de données n’existent pas déjà dans la base de données. |
| **Commit size**    | Saisissez le nombre de lignes de chaque lot à écrire sur le serveur MDM. |
### Use Transaction

Cochez cette case et, dans la liste **Component List**, sélectionnez une connexion existante à utiliser pour commiter la transaction.

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

### Global Variables

|------------------|---------------------------------------------------------------------------|

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant nécessite toujours un flux d’entrée afin de structurer des données en XML. Si vos données ne sont pas encore structurées XML, vous devez utiliser des composants tels que le <strong>tWriteXMLField</strong> afin de transformer ces données en XML. Pour plus d’informations concernant le <strong>tWriteXMLField</strong>, consultez tWriteXMLField à la page 4294. Vous pouvez augmenter le délai avant suspension de la connexion pour un Job utilisant ce composant afin de faciliter le traitement d’un grand volume d’enregistrements de données. Pour plus d’informations, consultez les paramètres d’exécution avancés pour les paramètres JVM sur Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>). Si vous utilisez un Job contenant le composant tMDMBulkLoad pour charger en masse de grands volumes de données dans MDM, vous pouvez personnaliser l’opération de chargement de masse en ajoutant un argument JVM spécifique (par exemple, bulkload.concurrent.http.requests=25) dans l’onglet <strong>Advanced settings</strong> du Job, afin de limiter le nombre maximum de requêtes simultanées envoyées au serveur MDM. Cet argument permet d’éviter de consommer toutes les connexions disponibles du</th>
</tr>
</thead>
</table>
serveur d’applications Tomcat, ce qui provoquera des problèmes de transaction et de blocage.

| Connections | Liens de sortie (de ce composant à un autre) :
| Row : Main. |

| Liens d’entrée (d’un autre composant à celui-ci) :
| Row : Main. |

Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.

Scénario : Charger des enregistrements dans une entité métier

Ce scénario s’applique uniquement aux solutions Talend avec MDM.

Ce scénario décrit un Job qui charge des enregistrements dans l’entité métier ProductFamily définie par un modèle de données spécifique dans le hub MDM.

Prérequis :

• Le conteneur de données Product. Ce conteneur de données est utilisé pour séparer le domaine de données maître des produits des autres domaines de données maître.

• Le modèle de données Product. Ce modèle de données est utilisé pour définir les attributs, les règles de validation, les droits d’accès des utilisateurs et les relations des entités en rapport. Il définit les attributs de l’entité métier ProductFamily.

• L’entité métier ProductFamily. Cette entité métier contient les attributs Id et Name, définis dans le modèle de données Product.

Pour plus d’informations concernant la création d’un conteneur de données, d’un modèle de données et d’une entité métier ainsi que de ses attributs, consultez la partie MDM de votre Guide utilisateur du Studio Talend.

Le Job dans ce scénario utilise trois composants.

• Un tFixedFlowInput. Ce composant génère les enregistrements à charger dans l’entité métier ProductFamily. Généralement, les enregistrements à charger sont volumineux et stockés dans un fichier spécifique. Cependant, pour faciliter la reproduction de ce scénario, ce Job utilise un tFixedFlowInput afin de générer quatre exemples d’enregistrements.
• Un **tWriteXMLField** : ce composant transforme les données d’entrée pour leur donner une structure XML.

• Un **tMDMBulkLoad** : ce composant écrit en masse les données d’entrée dans l’entité métier *ProductFamily* et génère une valeur d’ID pour chaque donnée de l’enregistrement.

**Placer et relier les composants**

**Procédure**

1. Déposez un **tFixedFlowInput**, un **tWriteXMLField** et un **tMDMBulkLoad** dans l’espace de modélisation graphique.
2. Reliez le **tFixedFlowInput** au **tWriteXMLField** à l’aide d’un lien **Main**.
3. Répétez l’opération pour relier le **tWriteXMLField** au **tMDMBulkLoad**.

**Configurer les composants**

**Générer les enregistrements de données à charger dans une entité métier**

**Procédure**

1. Double-cliquez sur le **tFixedFlowInput** afin d’ouvrir sa vue **Basic settings**.
2. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur du schéma.
3. Dans l’éditeur du schéma, cliquez sur le bouton [+] pour ajouter une colonne.
4. Nommez la nouvelle colonne *family* dans cet exemple.
5. Cliquez sur **OK** pour fermer l’éditeur du schéma.
6. Dans la partie **Mode** de la vue **Basic settings**, sélectionnez l’option **Use Inline Table**.
7. Cliquez quatre fois sur le bouton [+] pour ajouter quatre lignes à la table.
8. Dans le tableau, cliquez sur chacune des lignes ajoutées et saisissez leur nom, entre guillemets : *Shirts, Hats, Pets et Mugs* (Chemises, chapeaux, vêtements pour animaux, tasses).

**Transformer les données entrantes en structure XML**

**Procédure**

1. Double-cliquez sur le tWriteXMLField pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur du schéma puis ajoutez une ligne en cliquant sur le bouton [+].
3. Cliquez sur la nouvelle ligne, dans la partie droite de l’éditeur du schéma et saisissez le nom de la colonne de sortie dans laquelle vous souhaitez écrire le contenu XML. Dans cet exemple, saisissez xmlRecord.

4. Cliquez sur OK pour valider ce schéma de sortie et fermer l’éditeur du schéma.

Dans la boîte de dialogue qui s’ouvre, cliquez sur OK afin de propager le schéma au composant suivant.

5. Dans la vue Basic settings, cliquez sur le bouton [...] à côté du champ Configure Xml Tree pour ouvrir l’interface permettant de créer la structure XML.

6. Dans la zone Link Target, cliquez sur rootTag et renommez le champ ProductFamily, qui est le nom de l’entité métier utilisée dans ce scénario.

7. De la zone Linker source, glissez-déposez family dans ProductFamily, dans la zone Link target.

Une boîte de dialogue s’affiche et vous demande quel type d’opération vous souhaitez effectuer. Sélectionnez Create as sub-element of target node afin de créer un sous-élément du noeud ProductFamily. L’élément family apparaît sous le noeud ProductFamily.

Dans la zone Link target, cliquez sur le nœud family et renommez-le Name, qui est l’un des attributs de l’entité métier ProductFamily.

Cliquez-droit sur le nœud Name et sélectionnez Set As Loop Element dans le menu contextuel.

8. Dans la zone Link target, cliquez sur le nœud family et renommez-le Name, qui est l’un des attributs de l’entité métier ProductFamily.

Cliquez sur OK afin de valider la structure XML que vous avez définie.
Écrire les données entrantes dans l’entité métier

**Procédure**

1. Double-cliquez sur le tMDMBulkLoad afin d’ouvrir sa vue Basic settings.

![tMDMBulkLoad](image)

2. Dans la liste XML Field, sélectionnez xmlRecord.


4. Dans les champs Username et Password, saisissez respectivement votre identifiant et votre mot de passe de connexion au serveur MDM.

5. Dans les champs Data Model et Data Container, saisissez les noms correspondants au modèle de données et au conteneur de données que vous souhaitez utiliser, *Product* pour les deux champs, dans ce scénario.

   Dans le champ Entity, saisissez le nom de l’entité métier dont les enregistrements doivent être chargés. Dans cet exemple, saisissez *ProductFamily*.

6. Cochez la case Generate ID afin de générer des valeurs d’ID pour les enregistrements à charger.

7. Dans le champ Commit size, saisissez la taille du lot à écrire dans le hub MDM en mode bulk.

**Enregistrer et exécuter le Job**

**Procédure**

1. Appuyez sur Ctrl+S pour enregistrer votre Job.

2. Exécutez le Job en appuyant sur F6 ou en cliquant sur Run dans l’onglet Run.

   Connectez-vous à votre interface Talend MDM Web UI pour vérifier les nouveaux enregistrements ajoutés à l’entité métier ProductFamily.
tMDMClose

Ce composant ferme un connexion ouverte au serveur MDM après l’exécution du sous-job précédent.

Propriétés du tMDMClose Standard

Ces propriétés sont utilisées pour configurer le tMDMClose s’exécutant dans le framework de Jobs Standard.

Le composant tMDMClose Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

Basic settings

| Component List | S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tMDMConnection dans la liste. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant doit être utilisé avec le tMDMConnection. Notez que les composants tMDMCommit et tMDMRollback comprennent également une option, dans leur onglet Basic settings, permettant de fermer une connexion. |
Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion au serveur MDM parmi celles prévues dans votre Job.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénario associé

Pour un scénario associé, consultez tMDMDelete à la page 2262.
**tMDMCommit**

Ce composant committe toutes les modifications apportées à une base de données dans le cadre d’une transaction dans MDM.

Ce composant est utilisé pour contrôler le moment où, dans un Job MDM, les modifications apportées à la base de données sont commitées, dans le cadre d’une transaction MDM, par exemple pour éviter des commits partiels lorsqu’une erreur survient.

Pour plus d’informations concernant les transactions MDM, consultez l’article relatif aux transactions MDM sur Talend Help Center (https://help.talend.com).

**Propriétés du tMDMCommit Standard**

Ces propriétés sont utilisées pour configurer le tMDMCommit s’exécutant dans le framework de Jobs Standard.

Le composant tMDMCommit Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Component List</th>
<th>Sélectionnez le composant <strong>tMDMConnection</strong> pour lequel effectuer l’action de commit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close Connection</td>
<td>Cochez cette case pour fermer la session de la connexion au serveur MDM après avoir commité les modifications. Notez que, même si la case n’est pas cochée, la connexion ne peut toujours pas être utilisée par le sous-job suivant, sauf si le mode Auto-Commit est activé.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau des composants. |

**Global Variables**

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec les composants tMDM Connection, tMDMRollback, tMDMSP, tMDMViewSearch, tMDMInput, tMDMDelete, tMDMRouteRecord, tMDMOutput et tMDMClose. |

**Scénario associé**

Pour un scénario associé, consultez Scénario : Supprimer des données maître d’un hub MDM à la page 2265.
tMDMConnection

Ce composant ouvre une connexion au serveur MDM afin d’en permettre la réutilisation dans le Job ou la transaction en cours.

Propriétés du tMDMConnection Standard

Ces propriétés sont utilisées pour configurer le tMDMConnection s’exécutant dans le framework de Jobs Standard.

Le composant tMDMConnection Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

Basic settings

| MDM version | Par défaut, l’option Server 6.0 est sélectionnée. Bien qu’il soit recommandé de migrer les Jobs existants vers cette nouvelle version, l’option Server 5.6 est disponible pour faciliter le processus de migration de vos Jobs, afin qu’ils continuent à fonctionner avec un serveur 6.0 sans modification. Pour ce faire, une option sur le serveur doit être activée pour accepter et traduire les requêtes de ces Jobs. |
| URL | Saisissez l’URL pour accéder au serveur MDM. |
| Username et Password | Informations d’authentification de l’utilisateur du serveur MDM.

Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Auto Commit | Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Dans un contexte MDM, avec la fonction Auto-Commit activée, les requêtes envoyées au serveur MDM seront commitées après leur exécution. Cependant, si vous souhaitez gérer vos transactions au sein d’un Job avec plus de précision, il est recommandé d’utiliser le composant tMDMCommit. |
Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant doit être utilisé avec les composants tMDMCommit, tMDMRollback tMDMSP, tMDMViewSearch, tMDMInput, tMDMDelete, tMDMRouteRecord, tMDMOutput, et tMDMClose. |

Scénario associé

Pour un scénario associé, consultez tMDMDelete à la page 2262.
tMDMDelete

Ce composant supprime des enregistrements de données maître d'entités spécifiques du hub MDM.

Propriétés du tMDMDelete Standard

Ces propriétés sont utilisées pour configurer le tMDMDelete s'exécutant dans le framework de Jobs Standard.

Le composant tMDMDelete Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
Cliquez sur le bouton Sync columns afin de récupérer le schéma du composant précédent.


<p>| Use an existing connection | Cochez cette case et sélectionnez le composant tMDMConnection adéquat à partir de la liste Component pour réutiliser les paramètres d'une connexion que vous avez déjà définie. |</p>
<table>
<thead>
<tr>
<th>MDM version</th>
<th>Par défaut, l’option <strong>Server 6.0</strong> est sélectionnée. Bien qu’il soit recommandé de migrer les Jobs existants vers cette nouvelle version, l’option <strong>Server 5.6</strong> est disponible pour faciliter le processus de migration de vos Jobs, afin qu’ils continuent à fonctionner avec un serveur 6.0 sans modification. Pour ce faire, une option sur le serveur doit être activée pour accepter et traduire les requêtes de ces Jobs.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URL</strong></td>
<td>Saisissez l’URL d’accès au serveur MDM.</td>
</tr>
</tbody>
</table>
| **Username et Password** | Informations d’authentification de l’utilisateur du serveur MDM.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Entity**  | Saisissez le nom de l’entité contenant les données que vous souhaitez effacer.                   |
| **Data Container** | Saisissez le nom du conteneur de données contenant les données que vous souhaitez effacer. |
| **Type**    | Sélectionnez **Master** ou **Staging** pour spécifier sur quelle base de données effectuer l’action. |
| **Use multiple conditions** | Cochez cette case afin de filtrer les données maître à supprimer, en respectant certaines conditions.  
**Xpath** : Saisissez entre guillemets le chemin d’accès et le nœud XML auquel vous souhaitez appliquer la condition.  
**Function** : Sélectionnez dans la liste la condition à utiliser.  
**Value** : Saisissez entre guillemets la valeur à utiliser.  
**Predicate** : Sélectionnez un prédicat si vous utilisez plus d’une condition.  
Pour plus d’informations concernant les opérateurs et les prédicats, consultez **Propriétés du tMDMInput Standard** à la page 2272. |
| **Keys (in sequence order)** | Spécifiez le(s) champ(s) (dans l’ordre de la séquence) composant la clé lorsque l’entité possède une clé multiple. |
| **Fire Delete event** | **Remarque** :  
Cette case est disponible uniquement si vous sélectionnez l’option **Master** dans la liste **Type**.  
Cochez cette case pour qu’un événement Delete soit généré et écrit dans le rapport **UpdateReport**, qui fournit un rapport complet sur toutes les actions de création, de mise à jour ou de suppression dans les données maître. |
**Logical delete**

Cochez cette case afin d'envoyer les données maître dans la Corbeille, et renseignez le chemin d'accès à la Corbeille (*Recycle bin path*). Une fois dans la Corbeille, les données maître peuvent être supprimées définitivement ou restaurées. Si vous laissez la case décochée, les données maître seront supprimées définitivement.

**Die on error**

Cochez cette case pour passer la ligne en erreur et terminer le processus afin d'obtenir des lignes ne contenant aucune erreur. Si nécessaire, vous pouvez récupérer les lignes en erreur en utilisant un lien de type *Row > Rejects*.

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td>NB_LINE : nombre de lignes traitées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
</tbody>
</table>

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

**Règle d’utilisation**

Si la case *Use multiple conditions* est cochée, le *tMDMDelete* peut fonctionner en standalone. Sinon, il requiert un lien d’entrée.
Vous pouvez augmenter le délai avant suspension de la connexion pour un Job utilisant ce composant afin de faciliter le traitement d'un grand volume d'enregistrements de données. Pour plus d'informations, consultez les paramètres d'exécution avancés pour les paramètres JVM sur Talend Help Center (https://help.talend.com).

**Scénario : Supprimer des données maître d'un hub MDM**

Ce scénario s'applique uniquement aux solutions Talend avec MDM.

Ce scénario décrit un Job à six composants qui supprime un enregistrement de données spécifié du référentiel XML MDM. Il utilise les composants `tMDMCommit` et `tMDMRollback` afin de ne propager les modifications à la base de données uniquement lorsque le Job est correctement exécuté.

**Déposer et relier les composants**

**Procédure**

1. Déposez un composant `tMDMConnection`, un `tMDMCommit`, un `tMDMClose`, un `tMDMInput`, un `tMDMDelete` et un `tMDMRollback` de la famille Talend MDM de la Palette dans l'espace de modélisation graphique.
2. Reliez le `tMDMInput` au `tMDMDelete` à l'aide d'un lien `Row > Main`.
3. Reliez le composant `tMDMConnection` au `tMDMInput` à l'aide d'un lien `Trigger > OnSubjobOk`.
4. Reliez le `tMDMDelete` au `tMDMCommit`, le `tMDMCommit` au `tMDMClose` et le `tMDMRollback` au `tMDMClose` à l'aide de liens `Trigger > OnComponentOk`.
5. Reliez le `tMDMDelete` au `tMDMRollback` à l'aide d'un lien `Trigger > OnComponentError`. 
Configurer la connexion au serveur MDM

Pourquoi et quand exécuter cette tâche

Dans ce scénario, un composant **tMDMConnection** est utilisé pour ouvrir une connexion au serveur MDM et la réutiliser ultérieurement dans le sous-job suivant, qui supprime l’enregistrement de données spécifié.

Procédure

1. Double-cliquez sur le **tMDMConnection** afin d’afficher sa vue **Basic settings** et définir ses propriétés de base.


3. Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant et votre mot de passe pour vous connecter au serveur MDM.

4. Double-cliquez sur le **tMDMCommit** pour afficher sa vue **Basic settings**.

Ce composant commite les modifications apportées à la base de données lorsque le sous-job est exécuté avec succès.

5. Dans la liste **Component List**, sélectionnez le composant de connexion au serveur que vous souhaitez fermer, si vous avez configuré plusieurs connexions à des serveurs MDM. Dans ce scénario, une seule connexion au serveur est ouverte, utilisez donc simplement le paramètre par défaut.

6. Décochez la case **Close Connection** si elle est cochée. Dans cet exemple, le composant **tMDMClose** ferme la connexion au serveur MDM.
7. Double-cliquez sur le composant **tMDMRollback** pour afficher sa vue **Basic settings**.

Ce composant effectue un rollback des modifications et rétablit l’état précédent de la base de données si le sous-job échoue.

8. Dans la liste **Component List**, sélectionnez le composant de connexion au serveur que vous souhaitez fermer, si vous avez configuré plusieurs connexions à des serveurs MDM. Dans ce scénario, une seule connexion au serveur est ouverte, utilisez donc simplement le paramètre par défaut.

9. Décochez la case **Close Connection** si elle est cochée. Dans cet exemple, le composant **tMDMClose** ferme la connexion au serveur MDM.

10. Double-cliquez sur le composant **tMDMClose** pour afficher sa vue **Basic settings** et définir ses propriétés de base.

Le composant **tMDMClose** est utilisé pour fermer la connexion après l’exécution réussie du Job. Vous pouvez également fermer la connexion en cochant la case **Close Connection** dans les composants **tMDMCommit** et **tMDMRollback**. Dans un souci de démonstration, un **tMDMClose** est utilisé dans ce scénario.

11. Dans la liste **Component List**, sélectionnez le composant dont vous souhaitez fermer la connexion au serveur, si vous avez configuré plus d’une connexion au serveur MDM. Dans ce scénario, une seule connexion est ouverte, utilisez simplement la configuration par défaut.

**Configurer la récupération de données**

**Procédure**

1. Double-cliquez sur le **tMDMInput** pour afficher sa vue **Basic settings** et définir ses propriétés de base.
2. Dans la liste **Property Type**, sélectionnez **Built-in** afin de renseigner les champs manuellement. Si vous avez stocké vos informations de connexion au serveur MDM dans le référentiel des métadonnées, sélectionnez **Repository** dans la liste et les champs seront automatiquement renseignés.

3. Dans la liste **Schema**, sélectionnez **Built-in** et cliquez sur le bouton **[...]** à côté du champ **Edit schema** pour ouvrir une boîte de dialogue. Vous pouvez définir la structure des données maître que vous souhaitez lire dans le hub MDM.

4. Les données maître sont collectées dans trois colonnes de type **String** : Id, Name et Price. Cliquez sur **OK** pour fermer la boîte de dialogue et passer à l’étape suivante.

5. Cochez la case **Use an existing connection** et, dans la liste **Component List** qui apparaît, sélectionnez le composant que vous avez configuré pour ouvrir votre connexion au serveur MDM. Dans ce scénario, une seule connexion au serveur MDM existe, utilisez donc la sélection par défaut.
6. Dans le champ **Entity**, saisissez entre guillemets le nom de l’entité métier contenant les enregistrements de données que vous souhaitez lire. Ici, saisissez le nom de l’entité *Product*.

7. Dans le champ **Data Container**, saisissez entre guillemets le nom du conteneur de données contenant les données maître que vous souhaitez lire. Dans cet exemple, utilisez le conteneur *Product*.

   **Remarque :**
   La case **Use multiple conditions** est cochée par défaut.

8. Dans la table **Operations**, définissez comme suit les conditions pour filtrer les données maître que vous souhaitez supprimer :
   a) Cliquez sur le bouton "+" pour ajouter une nouvelle ligne.
   b) Dans la colonne **Xpath**, saisissez entre guillemets le chemin Xpath et la balise du nœud XML sur lequel appliquer le filtre. Dans cet exemple, vous travaillez avec l’entité *Agency*, saisissez donc "*Product/Name".
   c) Dans la colonne **Function**, sélectionnez la fonction que vous souhaitez utiliser. Dans ce scénario, utilisez la fonction *Contains*.
   d) Dans la colonne **Value**, saisissez la valeur de votre filtre. Filtrez les données maître dont le nom contient *mug*.

9. Dans la vue **Component**, cliquez sur **Advanced settings** afin de configurer les paramètres avancés.

10. Dans le champ **Loop XPath query**, saisissez entre guillemets la structure et le nom du nœud XML sur lequel la boucle doit être effectuée.

11. Dans la table **Mapping** et dans la colonne **XPath query**, saisissez entre guillemets le nom de la balise XML dans laquelle vous souhaitez collecter les données maître, à côté du nom de la colonne de sortie correspondante.

**Configurer la suppression de l’enregistrement de données**

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le composant *tMDMDelete* afin d’afficher sa vue **Basic settings** et définir ses propriétés de base.
2. Dans la liste **Schema**, sélectionnez **Built-in** puis cliquez sur le bouton [...] à côté du champ **Edit Schema** afin de décrire la structure des données maître dans le hub MDM.

3. Cliquez sur le bouton [+] de droite afin d’ajouter une colonne de type **String**. Nommez cette colonne **outputXML**. Cliquez sur **OK** pour fermer la boîte de dialogue et passer à l’étape suivante.

4. Cochez la case **Use an existing connection**, puis, dans la liste **Component List** qui apparaît, sélectionnez le composant que vous avez configuré pour ouvrir votre connexion au serveur MDM. Dans ce scénario, une seule connexion au serveur MDM existe, utilisez la sélection par défaut.

5. Dans le champ **Entity**, saisissez le nom de l’entité métier contenant les données maître que vous souhaitez supprimer, l’entité **Product** dans cet exemple.


   **Remarque :**
   Si l’entité possède plusieurs clés, ajoutez autant de ligne que nécessaire pour les clés et sélectionnez-les en ordre séquentiel.

8. Si nécessaire, cochez la case **Fire Delete event**, pour que l’événement de suppression soit généré et écrit dans le rapport **UpdateReport**.

9. Cochez la case **Logical delete** si vous ne souhaitez pas supprimer les données maître de manière permanente. Cela envoie les données supprimées dans la corbeille. Une fois dans la corbeille, les données maître peuvent être restaurées ou supprimées définitivement. Si vous laissez la case décochée, les données maître seront supprimées définitivement.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job et vous assurer que tous les paramètres configurés seront pris en compte.

2. Appuyez sur **F6** pour exécuter votre Job.

**Résultats**

Les données maître dont le nom contient "mug" ont été supprimées et envoyées dans la corbeille.
tMDMInput

Ce composant lit les données dans un hub MDM et rend possible le traitement de ces données.

Propriétés du tMDMInput Standard

Ces propriétés sont utilisées pour configurer le tMDMInput s’exécutant dans le framework de Jobs Standard.

Le composant tMDMInput Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].


### Use an existing connection
Cochez cette case et sélectionnez le composant **tMDMConnection** adéquat à partir de la liste **Component** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

### MDM version
Par défaut, l’option **Server 6.0** est sélectionnée. Bien qu’il soit recommandé de migrer les Jobs existants vers cette nouvelle version, l’option **Server 5.6** est disponible pour faciliter le processus de migration de vos Jobs, afin qu’ils continuent à fonctionner avec un serveur 6.0 sans modification. Pour ce faire, une option sur le serveur doit être activée pour accepter et traduire les requêtes de ces Jobs.

### URL
Saisissez l’URL d’accès au serveur MDM.

### Username et Password
Saisissez les informations d’authentification de l’utilisateur du serveur MDM.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

### Entity
Saisissez le nom de l’entité contenant les données que vous souhaitez effacer.

### Data Container
Saisissez le nom du conteneur de données contenant les données que vous souhaitez effacer.

### Type
Sélectionnez **Master** ou **Staging** pour spécifier sur quelle base de données effectuer l’action.

### Use multiple conditions
Cochez cette case pour filtrer les données à l’aide de conditions.

**Xpath** : saisissez, entre guillemets, le nœud XML sur lequel appliquer la condition.

**Function** : sélectionnez la condition à utiliser.

Avant d’utiliser les conditions, ayez ceci à l’esprit :

- selon le type de champ vers lequel pointe le XPath, seuls certains opérateurs s’appliquent. Par exemple, si le champ est un booléen, seuls les opérateurs **Equal** et **Not Equal** s’appliquent.

- L’opérateur **Not Equal** ne supporte pas les champs multi-occurrence ou les champs de type complexe.

Les opérateurs suivants sont disponibles :

- **Contains** : Retourne un résultat contenant le ou les mot(s) saisi(s). Notez que la rechercher plein texte ne supporte pas les caractères spéciaux, par exemple @, #, $.

- **Contains the sentence** : Retourne un ou plusieurs résultat(s) contenant la séquence saisie.
- **Joins With** : Cet opérateur est réservé pour un usage ultérieur.

- **Starts With** : Retourne un résultat commençant par la chaîne de caractères saisie.

- **Equal** : Retourne un résultat correspondant à la valeur saisie.

- **Not Equal** : Retourne un résultat dont la valeur n’est pas null et est différente de la valeur saisie.

- **is greater than** : Retourne un résultat supérieur à la valeur numérique saisie. S’applique uniquement aux champs contenant des nombres.

- **is greater or equal** : Retourne un résultat supérieur ou égal à la valeur numérique saisie. S’applique uniquement aux champs contenant des nombres.

- **is lower than** : Retourne un résultat inférieur à la valeur numérique saisie. S’applique uniquement aux champs contenant des nombres.

- **is lower or equal** : Retourne un résultat inférieur ou égal à la valeur numérique saisie. S’applique uniquement aux champs contenant des nombres.

- **whole content contains** : Effectue une recherche plein texte utilisant le champ spécifié Xpath dans le conteneur de données sélectionné. Si vous saisissez une chaîne de caractères vide "" dans le champ Xpath et que vous sélectionnez **whole content contains** dans la liste **Function**, les recherches seront effectuées dans tous les champs de toutes les entités du conteneur de données sélectionné.

- **is empty or null** : Retourne un champ vide ou une valeur nulle.

**Value** : saisissez, entre guillemets, la valeur souhaitée. Notez que si la valeur contient des caractères XML spéciaux, comme /, vous devez saisir la valeur entre guillemets simples également ("'ABC/XYZ'"), sinon la valeur est considérée comme XPath.

**Predicate** : sélectionnez un prédicat lorsque vous utilisez plusieurs conditions.

Les prédicats suivants sont disponibles :

- **Default** : Interpréte comme un and.

- **or** : L'une des conditions s’applique.

- **and** : Toutes les conditions s’appliquent.

Les autres prédicats sont réservés pour usage ultérieur et peuvent être sujets à des comportements inattendus.

Si vous décochez cette case, vous avez la possibilité de sélectionner des ID particuliers devant être affichés dans la colonne **ID value** du tableau **IDS**.

**Remarque** :
Si vous décochez la case **Use multiple conditions**, l’option **Batch Size** de l’onglet **Advanced settings** ne sera plus disponible.

| Skip Rows | Renseignez le nombre de lignes à ignorer. |
| Max Rows | Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n’est lue ou traitée. |
| Die on error | Cochez cette case pour passer la ligne en erreur et terminer le processus afin d’obtenir des lignes ne contenant aucune erreur. Si nécessaire, vous pouvez récupérer les lignes en erreur en utilisant un lien de type **Row > Rejects**. |

### Advanced settings

| Batch Size | Nombre de lignes dans chaque lot traité. |
| Remarque : | Cette option n’apparaît pas si vous avez décoché la case **Use multiple conditions** dans l’onglet **Basic settings**. |
| Loop XPath query | Nœud de l’arborescence XML sur lequel est basée la boucle. |
| Mapping | **Column** : reflète le schéma tel qu’il est défini dans l’éditeur **Edit Schema**. |
| | **XPath query** : Saisissez les champs à extraire de la structure XML d’entrée. |
| | **Get Nodes** : Cochez cette case pour récupérer les balises XML entourant les données avec celles-ci. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
| **ERROR_MESSAGE** | message d’erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
Lire les données maître depuis un hub MDM

Ce scénario s’applique uniquement aux solutions Talend avec MDM.

Ce scénario décrit un Job à deux composants récupérant des données maître d’un serveur MDM et affichant ces données dans la console.

Prérequis :
- Vérifiez que le serveur MDM est en cours de fonctionnement.
- Vous avez importé le projet démo MDM et chargé les données d’exemple dans le conteneur de données Product, en exécutant le Job MDM_LoadAll.

Créer un Job lisant les données maître depuis MDM

Procédure
1. De la Palette, déposez un tMDMInput et un tLogRow dans l’espace de modélisation graphique.
2. Reliez les deux composants à l’aide d’un lien Row > Main.

Configurer les paramètres simples du tMDMInput pour lire les données maître depuis MDM

Procédure
1. Double-cliquez sur le tMDMInput pour ouvrir sa vue Basic settings.
2. Dans la liste **Property Type**, sélectionnez **Built-In**.

3. Dans la liste **Schema**, sélectionnez **Built-In** et cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir une fenêtre et définir la structure des données maître à lire depuis le serveur MDM.

Dans cet exemple, trois colonnes sont définies pour récupérer trois éléments de l’entité **Product** : **Name**, **Price** et **Colors**.

4. Une fois le schéma défini, cliquez sur **OK** pour fermer cette fenêtre, puis cliquez sur **Yes** dans la boîte de dialogue **[Propagate]**, afin de propager les modifications du schéma au **tLogRow**.

5. Saisissez l’utilisateur et le mot de passe pour accéder au serveur MDM.

6. Dans le champ **Entity**, saisissez **Product** entre guillemets.

7. Dans le champ **Data Container**, saisissez **Product** entre guillemets.

8. Sélectionnez **Master** dans la liste **Type**.

9. Définissez les conditions de requête dans la zone **Operations**.
Dans cet exemple, vous allez effectuer une requête sur les enregistrements de données produits dont le nom contient *Shirt*.

a) Cliquez sur le bouton [+] pour ajouter une ligne.

b) Saisissez *Product/Name* entre guillemets dans le champ Xpath.

En plus des éléments définis dans les entités, vous pouvez effectuer des requêtes sur des éléments de métadonnées, également connus comme éléments built-in. Pour effectuer des requêtes sur des éléments de métadonnées à partir d’enregistrements dans la base de données maître, suivez le format `metadata:<timestamp|task_id>` lors de la définition du chemin, afin de sélectionner le nœud XML sur lequel exécuter la requête.

c) Sélectionnez **Contains** dans la liste **Function**.

d) Saisissez *Shirt*, entre guillemets, dans le champ **Value**.

**Configurer les paramètres avancés du tMDMInput pour lire les données maître depuis MDM**

**Procédure**

1. Dans la vue **Component**, cliquez sur l’onglet **Advanced settings**.

   ![tMDMInput_1](Image)

2. Dans le champ **Loop XPath query**, saisissez `/Product` entre guillemets, nœud sur lequel baser la boucle.

3. Dans la colonne **XPath query** de la table **Mapping**, saisissez le nom du nœud XML duquel collecter les données maître, à côté du nom de la colonne de sortie correspondante.

4. Cochez la case **Get Nodes** dans la ligne **Colors**, afin de récupérer le nœud XML avec ses données.

**Configurer l’affichage des données et exécuter le Job**

**Procédure**

1. Double-cliquez sur le **tLogRow** pour afficher sa vue **Basic settings**.

2. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour une lisibilité optimale des résultats.


Les enregistrements de données *Product* dont le nom comprend *Shirt* sont affichés dans la console, avec les valeurs des trois colonnes spécifiées.
<table>
<thead>
<tr>
<th>Name</th>
<th>Price</th>
<th>Colors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Talend Dog T-Shirt</td>
<td>15.99</td>
<td>&lt;Colors&gt;&lt;Color&gt;White&lt;/Color&gt;&lt;Color&gt;Light Pink&lt;/Color&gt;&lt;/Colors&gt;</td>
</tr>
<tr>
<td>Talend Golf Shirt</td>
<td>16.99</td>
<td>&lt;Colors&gt;&lt;Color&gt;White&lt;/Color&gt;&lt;Color&gt;Blue&lt;/Color&gt;&lt;/Colors&gt;</td>
</tr>
<tr>
<td>Talend Fitted T-Shirt</td>
<td>15.99</td>
<td>&lt;Colors&gt;&lt;Color&gt;White&lt;/Color&gt;&lt;Color&gt;Light Blue&lt;/Color&gt;&lt;/Colors&gt;</td>
</tr>
</tbody>
</table>
tMDMOutput

Ce composant écrit ou de supprime des données sur le serveur MDM.

Le composant tMDMOutput reçoit des données du composant qui le précède, génère un document XML, puis écrit des données dans un hub MDM en utilisant un champ de sortie.

Propriétés du tMDMOutput Standard

Ces propriétés sont utilisées pour configurer le tMDMOutput s’exécutant dans le framework de Jobs Standard.

Le composant tMDMOutput Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
</tbody>
</table>

Input Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

- Repository : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets.

### Build the document
Cochez cette case si vous souhaitez construire le document à partir d'un schéma plat. Dans ce cas, double-cliquez sur le composant et mappez votre schéma dans la boîte de dialogue qui s'ouvre.

Si la case est décochée, vous devez sélectionner, dans la liste **Predefined XML document**, une colonne de votre schéma contenant le document.

### Result of the XML serialization
 Sélectionnez le nom de la colonne de sortie dans laquelle vous souhaitez écrire les données XML.

### Use an existing connection
Cochez cette case pour utiliser un composant **tMDMConnection** déjà configuré.

### MDM version
Par défaut, l'option **Server 6.0** est sélectionnée. Bien qu'il soit recommandé de migrer les Jobs existants vers cette nouvelle version, l'option **Server 5.6** est disponible pour faciliter le processus de migration de vos Jobs, afin qu'ils continuent à fonctionner avec un serveur 6.0 sans modification.

Pour ce faire, une option sur le serveur doit être activée pour accepter et traduire les requêtes de ces Jobs.

### URL
Saisissez l'URL d'accès au serveur MDM.

### Username et Password
Saisissez les informations d'authentification de l'utilisateur du serveur MDM.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Remarque :**
Cet utilisateur doit avoir le bon rôle dans la solution MDM de Talend, c'est-à-dire qu'il puisse se connecter via un Job ou tout autre appel de service Web. Pour plus d'informations, consultez le Guide utilisateur du Studio Talend.

### Data Model
Saisissez le nom du modèle de données en comparaison duquel les données à écrire seront validées.

### Data Container
Saisissez le nom du conteneur de données dans lequel vous souhaitez écrire les données maître.

**Remarque :**
Ce conteneur de données doit avoir été créé.

### Type
Sélectionnez **Master** ou **Staging** pour spécifier sur quelle base de données effectuer l'action.

### Return Keys
**Columns corresponding to IDs in order** : dans l’ordre séquentiel, définissez les colonnes de sortie qui...
stockeront les valeurs des clés retournées (clés primaires) des éléments à créer.

<table>
<thead>
<tr>
<th>Is Update</th>
<th>Cochez cette case pour mettre à jour les champs modifiés. Si vous ne cochez pas cette case, la totalité des champs sera remplacée par les champs modifiés.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fire Create/Update event</td>
<td>Cochez cette case pour ajouter les actions effectuées à un rapport de modifications. <strong>Source Name</strong> : Saisissez, entre guillemets, le nom de l’application à partir de laquelle les modifications sont apportées. <strong>Enable verification by &quot;before saving&quot; transformer</strong> : Cochez cette case pour vérifier l’enregistrement qui vient d’être ajouté avant qu’il ne soit enregistré.</td>
</tr>
</tbody>
</table>
| Use partial update                                                       | Cochez cette case si vous devez mettre à jour des éléments multi-occurences (attributs) d’une entité existante à partir du contenu d’un flux XML d’entrée. Une fois la case cochée, vous devez configurer les paramètres ci-dessous : **Pivot** : saisissez le chemin XPath vers les sous-éléments multi-occurences dans lesquels des données doivent être ajoutées, remplacées ou supprimées dans l’entité de votre choix. Par exemple, vous souhaitez ajouter un sous-élément fils à une entité existante, comme suit :

```xml
<Person>
  <Id>1</Id> <!-- record key is mandatory -->
  <Children>
    <Child>[1234]</Child> <!-- FK to a Person Entity -->
  </Children>
</Person>
```

le chemin XPath saisi dans le champ **Pivot** doit être comme suit : /Person/Addresses/Adress la case **Overwrite** doit être cochée et la champ **Key** configuré comme suit : /Type.

Dans cet exemple, nous supposons que l’entité MDM a seulement une adresse de type *office*, ainsi cette adresse de type *office* sera remplacée et l’adresse de type *home* sera ajoutée.

- **Overwrite** : cochez cette case si vous devez remplacer ou mettre à jour les sous-éléments originaux par les sous-éléments d’entrée. Laissez cette case décochée si vous souhaitez ajouter un sous-élément.

- **Key** : saisissez le chemin XPath relatif au pivot qui permettra de mettre en correspondance le sous-élément du flux XML d’entrée avec le sous-élément de l’entité. Si une clé n’est pas fournie, tous les sous-éléments de l’entité ayant un XPath qui correspond à celui du sous-élément du flux XML d’entrée seront remplacés. Si
plus d'un sous-élément correspond à la clé, MDM met à jour le premier qu'il rencontre. Si aucun sous-élément ne correspond à la clé, elle est ajoutée à la fin de la collection.

- **Position** : saisissez un chiffre pour indiquer la position après laquelle les nouveaux éléments (ceux qui ne correspondent pas à la clé) seront ajouté. Si vous ne renseignez pas ce champ, les nouveau éléments seront ajoutés à la fin.

- **Delete** : cochez cette case si vous devez supprimer un ou plusieurs sous-éléments des sous-éléments originaux.

Par exemple, si vous devez supprimer deux "houses" de l'élément suivant :

```xml
<Person>
  <Id>1</Id>
  <Name>p1</Name>
  <Houses>
    <House>[1]</House>
    <House>[3]</House>
  </Houses>
  <Children>
    <Child>
      <Name>k1</Name>
      <Age>1</Age>
      <Habits>
        <Habit>Basketball</Habit>
        <Habit>Football</Habit>
        <Habit>Tennis</Habit>
        <Habit>Boxing</Habit>
      </Habits>
    </Child>
    <Child>
      <Name>k2</Name>
      <Age>2</Age>
    </Child>
  </Children>
</Person>
```

Le Xpath saisi dans le champ **Pivot** doit se présenter comme suit : `/Person/Houses/House`. La case **Delete** est cochée et le champ **Key** est configuré à `empty`. De plus, vous devez fournir le flux XML source comme suit :

```xml
<Person>
  <Id>1</Id>
  <Houses>
    <House>[1]</House>
  </Houses>
</Person>
```


Pour plus d'exemples relatifs à des opérations de mises à jour partielles, consultez **Exemples d'opérations de mises à jour partielles à l'aide du tMDMOutput** à la page 2285.
### Advanced settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Die on error</strong></td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Rejects.</td>
</tr>
</tbody>
</table>
| **Extended Output**             | Cochez cette case pour commiter les données maître par lot.  
Rows to commit : saisissez le nombre de lignes par lot. |
| **Configure Xml Tree**          | Ouvrez l’interface d’aide à la création de la structure XML des données maître à écrire.                                                   |
| **Group by**                    | Sélectionnez la colonne à utiliser pour regrouper les données maître.                                                                         |
| **Create empty element if needed** | Cette case est cochée par défaut. Si le contenu de la colonne Related Column de l'interface permettant de créer la structure XML est nul, ou si aucune colonne n’est associée au nœud XML, cette option créera une balise ouvrante et une balise fermante aux endroits prévus. |
| **Advanced separator (for number)** | Cochez cette case pour modifier les séparateurs utilisés par défaut dans les nombres :  
- Thousands separator : saisissez entre guillemets le séparateur à utiliser pour les milliers.  
- Decimal separator : saisissez entre guillemets le séparateur à utiliser pour les décimales. |
| **Generation mode**             | Sélectionnez le mode de génération correspondant à votre mémoire disponible. Les modes disponibles sont :  
- Lent et consommateur de mémoire (Slow and memory-consuming - Dom4J).  
  Remarque :  
  Cette option vous permet d’utiliser Dom4J pour traiter des fichiers XML très complexes.  
- Rapide et peu consommateur de mémoire (Fast with low memory consumption (SAX)). |
| **Encoding**                    | Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données. |
| **tStatCatcher Statistics**     | Cochez cette case pour collecter les données de log au niveau du composant.                                                                   |
Global Variables

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable *After* et retourne un entier.

**NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable *After* et retourne un entier.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

Utilisation

**Règle d’utilisation**

Utilisez ce composant pour écrire un fichier et séparer les champs à l’aide d’un séparateur spécifique.

Vous pouvez augmenter le délai avant suspension de la connexion pour un *Job* utilisant ce composant afin de faciliter le traitement d’un grand volume d’enregistrements de données. Pour plus d’informations, consultez les paramètres d’exécution avancés pour les paramètres JVM sur *Talend Help Center* (https://help.talend.com).

Exemples d’opérations de mises à jour partielles à l’aide du tMDMOutput

Cette section démontre comment configurer les paramètres dans le composant *tMDMOutput* pour effectuer des opérations de mises à jour partielles, via différents exemples.

**Mise à jour partielle avec écrasement activé ou désactivé**

Les deux exemples suivants expliquent comment ajouter des éléments à une entité existante avec des attributs multi-occurrence et mettre à jour les nouveaux éléments en se basant sur le contenu d’un flux XML source.

Voici un exemple d’élément existant :

```xml
<Person>
  <Id>1</Id>
  <Name>p1</Name>
  <Houses>
    <House>[1]</House>
```
Exemple 1 : Si vous souhaitez ajouter deux éléments Child à l’élément existant, le Xpath que vous saisissez dans le champ Pivot doit se présenter comme suit : /Person/Children/Child où la case Overwrite est décochée et le champ Key est configuré à /Name. Notez que l’élément Child est de type complexe et doit être identifié avec le champ Key. De plus, vous devez fournir le flux XML source comme suit :

```
<Person>
  <Id>1</Id>
  <Children>
    <Child>
      <Name>k2</Name>
    </Child>
    <Child>
      <Name>k3</Name>
    </Child>
  </Children>
</Person>
```

Dans ce cas, les deux éléments enfants Child [k2] et Child [k3] sont ajoutés et vous obtenez le résultat suivant :

```
<Person>
  <Id>1</Id>
  <Name>p1</Name>
  <Houses>
    <House>[1]</House>
    <House>[3]</House>
  </Houses>
  <Children>
    <Child>
      <Name>k1</Name>
      <Age>1</Age>
      <Habits>
        <Habit>Basketball</Habit>
        <Habit>Football</Habit>
        <Habit>Tennis</Habit>
        <Habit>Boxing</Habit>
      </Habits>
    </Child>
    <Child>
      <Name>k2</Name>
    </Child>
    <Child>
      <Name>k3</Name>
    </Child>
  </Children>
</Person>
```
Exemple 2 : Si vous souhaitez modifier le nom des deux éléments enfants *Child [k2]* et *Child [k3]*, l’expression XPath que vous saisissez dans le champ **Pivot** doit se présenter comme suit : `/Person/Children/Child` où la case **Overwrite** est cochée et le champ **Key** est configuré à `/Name`. De plus, vous devez fournir le flux XML source comme suit :

```xml
<Person>
  <Id>1</Id>
  <Children>
    <Child>
      <Name>Tina</Name>
    </Child>
    <Child>
      <Name>Tommy</Name>
    </Child>
  </Children>
</Person>
```

Dans ce cas, le nom des deux éléments enfants *Child [k2]* et *Child [k3]* est mis à jour et vous obtenez le résultat suivant :

```xml
<Person>
  <Id>1</Id>
  <Name>p1</Name>
  <Houses>
    <House>[1]</House>
    <House>[3]</House>
  </Houses>
  <Children>
    <Child>
      <Name>k1</Name>
      <Age>1</Age>
      <Habits>
        <Habit>Basketball</Habit>
        <Habit>Football</Habit>
        <Habit>Tennis</Habit>
        <Habit>Boxing</Habit>
      </Habits>
    </Child>
    <Child>
      <Name>Tina</Name>
    </Child>
    <Child>
      <Name>Tommy</Name>
    </Child>
  </Children>
</Person>
```

**Mise à jour partielle avec suppression activée**

Les exemples suivants expliquent comment supprimer certains éléments d’une entité existante avec des attributs multi-occurrence se basant sur le contenu d’un flux XML source.

Voici un exemple d’élément existant :

```xml
<Person>
  <Id>1</Id>
  <Name>p1</Name>
```
Exemple 1 : Si vous souhaitez supprimer deux éléments enfants de l’élément existant, l’expression Xpath que vous saisissez dans le champ Pivot doit se présenter comme suit : 

```
<Person/Children/Child
```

où la case Delete est cochée et le champ Key est configuré à /Name. Notez que l’élément Child est de type complexe et doit être identifié par son champ Key. De plus, vous devez fournir le flux XML source comme suit :

```
<Person>
  <Id>1</Id>
  <Children>
    <Child>
      <Name>k2</Name>
    </Child>
    <Child>
      <Name>k3</Name>
    </Child>
  </Children>
</Person>
```

La ligne XML source doit être affectée à la case UpdXML. Lorsqu'un élément est numéroté, le flux XML source peut être affecté à la case UpdXML sans affectation de la case UpdXML.
Dans ce cas, les deux éléments Child [k2] et Child [k3] sont supprimés et vous obtenez le résultat suivant :

```xml
<Person>
  <Id>1</Id>
  <Name>p1</Name>
  ...
  <Children>
    <Child>
      <Name>k1</Name>
      <Age>1</Age>
      <Habits>
        <Habit>Basketball</Habit>
        <Habit>Football</Habit>
        <Habit>Tennis</Habit>
        <Habit>Boxing</Habit>
      </Habits>
    </Child>
    ...
  </Children>
  ...
</Person>
```

**Exemple 2** : Si vous souhaitez supprimer des éléments habits pour l’élément Person, l’expression Xpath que vous saisissez dans le champ Pivot doit se présenter comme suit : 

```
/Person/Habits/Habit
```

où la case Delete est cochée et le champ Key est configuré à . ou empty. De plus, vous devez fournir le flux XML source comme suit :

```xml
<Person>
  <Id>1</Id>
  <Name>p1</Name>
  ...
  <Habits>
    <Habit>Basketball</Habit>
    <Habit>Football</Habit>
    <Habit>Boxing</Habit>
  </Habits>
</Person>
```

Dans ce cas, les trois éléments Habit [Basketball], Habit [Football] et Habit [Boxing] sont supprimés et vous obtenez le résultat suivant :

```xml
<Person>
  <Id>1</Id>
  <Name>p1</Name>
  ...
  <Habits>
    <Habit>Tennis</Habit>
  </Habits>
  ...
</Person>
```

**Exemple 3** : Si vous souhaitez supprimer des éléments habits à l’élément Child [k1], le Xpath que vous devez saisir dans le champ Pivot doit se présenter comme suit :

```
/Person/Children/Child[1]/Habits/Habit
```

où la case Delete est cochée et le champ Key est configuré à . ou empty. Notez que l’élément Child[1] est utilisé dans le champ Pivot pour identifier les habitudes (habits) de l’enfant (child) à supprimer. De plus, vous devez fournir le flux XML source comme suit :

```xml
<Person>
  <Id>1</Id>
  <Children>
    <Child>
      <Habits>
        <Habit>Basketball</Habit>
        <Habit>Football</Habit>
      </Habits>
    </Child>
  ...
</Person>
```
Dans ce cas, les deux éléments *Habit [Basketball]* et *Habit [Football]* sont supprimés et vous obtenez le résultat suivant :

```xml
<Person>
  <Id>1</Id>
  <Name>p1</Name>
  ...
  <Children>
    <Child>
      <Name>k1</Name>
      <Age>1</Age>
      <Habits>
        <Habit>Tennis</Habit>
        <Habit>Boxing</Habit>
      </Habits>
    </Child>
    <Child>
      <Name>k2</Name>
      <Age>2</Age>
    </Child>
    <Child>
      <Name>k3</Name>
      <Age>3</Age>
    </Child>
  </Children>
  ...
</Person>
```

**Exemple 4** : Si vous souhaitez supprimer des attributs *pets* (animaux de compagnie) pour un élément *Person*, l’expression Xpath que vous saisissez dans le champ *Pivot* doit se présenter comme suit : `/Person/Pets` où la case *Delete* est cochée et le champ *Key* est configuré à `/Pet`. De plus, vous devez fournir le flux XML source comme suit :

```xml
<Person>
  <Id>1</Id>
  <Pets>
    <Pet>Cat</Pet>
  </Pets>
  <Pets>
    <Pet>Pig</Pet>
  </Pets>
</Person>
```

Dans ce cas, les deux éléments *Pet [Cat]* et *Pet [Pig]* sont supprimés et vous obtenez le résultat suivant :

```xml
<Person>
  <Id>1</Id>
  <Name>p1</Name>
  ...
  <Pets>
    <Pet>Dog</Pet>
  </Pets>
  <Pets>
    <Pet>Cow</Pet>
  </Pets>
</Person>
```
Scénario : Écrire des données maître dans un hub MDM

Ce scénario s’applique uniquement aux solutions Talend avec MDM.

Ce scénario décrit un Job à deux composants qui génère un enregistrement de données, transforme ces données en XML et les charge dans l’entité métier définie sur le serveur MDM.

Dans cet exemple, l’objectif est de charger une nouvelle agence dans l’entité métier Agency. Cette nouvelle agence possède un identifiant, un nom et trois bureaux situés dans différentes villes.

Pour plus d’informations sur les entités, consultez le Guide utilisateur du Studio Talend.

Placer et relier les composants

Procédure
1. À partir de la Palette, glissez les composants tFixedFlowInput et tMDMOutput dans l’espace de modélisation graphique.
2. Reliez-les via un lien de type Row > Main.

Configurer les composants

Préparer le chargement des données dans le serveur MDM

Procédure
1. Double-cliquez sur le composant tFixedFlowInput pour afficher l’onglet Basic settings de la vue Component.
2. Dans la liste **Schema**, sélectionnez **Built-In** et cliquez sur le bouton [...] à côté du champ **Edit schema** pour décrire la structure des données maître que vous souhaitez écrire dans le serveur MDM.

3. Ajoutez cinq colonnes de type **String** en cliquant sur le bouton [+].
   Pour ce scénario, nommez-les **Id**, **Name**, **Office_R_and_D**, **Office_Sales**, et **Office_Services**.
4. Cliquez sur **OK** pour valider vos modifications.
5. Dans le champ **Number of rows**, saisissez le nombre de lignes que vous souhaitez générer.
6. Dans la zone **Mode**, sélectionnez le mode **Use Single Table** pour ne générer qu’une seule table.

**Paramètres simples du tMDMOutput**

**Procédure**

1. Dans l’espace de modélisation graphique, cliquez sur le **tMDMOutput** pour ouvrir sa vue **Basic settings**.
2. Dans la liste Schema, sélectionnez Built-In et cliquez sur le bouton […” à côté du champ Edit schema pour décrire la structure des données maître que vous souhaitez charger sur le serveur MDM.

Le composant tMDMOutput génère un document XML, l’écrit dans un champ de sortie et l’envoie au serveur MDM.

3. Cliquez sur OK pour passer à l’étape suivante.

La liste Result of the XML serialization dans l’onglet Basic settings est automatiquement renseignée avec la colonne xml.

4. Dans le champ URL, saisissez entre guillemets l’URL de connexion au serveur MDM.

5. Dans les champs Username et Password, saisissez l’identifiant et le mot de passe de connexion au serveur MDM.

6. Dans le champ Data Model, saisissez entre guillemets le nom du modèle de données par rapport auquel vous souhaitez valider les données maître que vous souhaitez écrire.
7. Dans le champ **Data Container**, saisissez entre guillemets le nom du conteneur de données dans lequel les données maître doivent être écrites.

8. Cochez la case **Is Update** si vous souhaitez uniquement mettre à jour des champs et non l’enregistrement de données complet.

**Paramètres avancés du tMDMOutput**

**Procédure**

1. Dans la vue **Component**, cliquez sur l’onglet **Advanced settings** pour configurer les paramètres avancés du composant.

2. Cochez la case **Extended Output** si vous souhaitez commiter les données maître par lot. Vous pouvez spécifier le nombre de lignes par lot dans le champ **Rows to commit**.

   Vous pouvez également double-cliquer sur le **tMDMOutput** pour ouvrir son éditeur.

3. Dans la zone **Link target** à droite, cliquez dans le champ **Xml Tree** et remplacez **rootTag** par le nom de l’entité métier dans laquelle vous souhaitez insérer l’enregistrement de données, **Agency** dans cet exemple.
Dans la zone **Linker source**, sélectionnez les deux colonnes de votre schéma *Id* et *Name* et glissez-les sur le nœud *Agency*.
La boîte de dialogue [Selection] s’ouvre.
Sélectionnez *Create as sub-element of target node* pour que vos deux colonnes correspondent à deux sous-éléments du nœud *Agency* en XML. Cliquez sur OK.

5. Cliquez-droit sur le nœud racine *Agency* puis sélectionnez *Add Sub-element*.
Dans la boîte de dialogue qui s’ouvre, saisissez le nom du nouveau sous-élément, *Offices* dans cet exemple.
Répétez la même procédure pour créer trois nouveaux sous-éléments *Office* du nœud *Offices* qui correspond à l’élément multi-occurrences *Offices* de l’entité métier *Agency*.

6. Dans la zone **Linker source**, sélectionnez les trois colonnes du schéma *Office_R_and_D*, *Office_Sales* et *Office_Services* et glissez-les respectivement dans les trois nouveaux nœuds *Office*.
La boîte de dialogue [Selection] s’ouvre.
Sélectionnez *Create as sub-element of target node* pour que les trois colonnes correspondent aux trois sous-éléments du nœud *Offices* en XML.

7. Cliquez sur **Ok** pour passer à l’étape suivante

8. Dans la zone **Link target** cliquez-droit sur l’élément que vous souhaitez définir comme élément sur lequel effectuer une boucle, puis sélectionnez *Set as Loop Element* dans le menu contextuel.
Dans cet exemple, *City* est l’élément d’itération.


### Enregistrer et exécuter le Job

**Procédure**

1. Appuyez sur Ctrl+S pour enregistrer votre Job.
2. Exécutez le Job en appuyant sur F6 ou en cliquant sur Run dans l’onglet Run.
Le nouvel enregistrement de données est inséré dans l’entité métier Agency, dans le conteneur de données DStar, sur le serveur MDM. Cet enregistrement de données contient, comme défini dans le schéma, l’ID de l’agence, son nom et ses trois bureaux situés dans trois villes.

**Master Data Container Browser DStar 0.1**

<table>
<thead>
<tr>
<th>Date</th>
<th>Entity</th>
<th>Keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>20160120 13:53:29</td>
<td>Agency</td>
<td>UT05</td>
</tr>
<tr>
<td>20150120 13:53:58</td>
<td>Agency</td>
<td>NV01</td>
</tr>
<tr>
<td>20160120 14:23:15</td>
<td>Agency</td>
<td>PA05</td>
</tr>
</tbody>
</table>

**Supprimer partiellement des données maître du hub MDM**

Ce scénario s'applique uniquement aux solutions Talend avec MDM.

Ce scénario explique comment supprimer partiellement les données maître écrites dans le serveur MDM dans le scénario Scénario : Écrire des données maître dans un hub MDM à la page 2291.
Dans cet exemple, un bureau de l’agence va être supprimé de l’entité métier *Agency*. L’agence possède actuellement un identifiant, un nom et trois bureaux situés dans différentes villes.

Pour plus d’informations sur les entités, consultez le *Guide utilisateur du Studio Talend*.

### Placer et relier les composants

**Procédure**

1. À partir de la Palette, glissez les composants *tFixedFlowInput* et *tMDMOutput* dans l’espace de modélisation.
2. Reliez-les via un lien de type *Row > Main*.

### Configurer les composants

**Spécifiez les données à supprimer du serveur MDM**

**Procédure**

1. Double-cliquez sur le composant *tFixedFlowInput* pour afficher l’onglet *Basic settings* de la vue *Component*. 

```plaintext
<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>&quot;PA05&quot;</td>
</tr>
</tbody>
</table>
| Name         | "Talend"
| Remove.Office| "Paris"
```

2. Enregistrez les modifications.
2. Dans la liste Schema, sélectionnez Built-In et cliquez sur le bouton [...] à côté du champ Edit schema pour ouvrir une boîte de dialogue dans laquelle vous pouvez définir la structure des données maître à utiliser pour supprimer partiellement les données maître du serveur MDM.

3. Ajoutez trois colonnes de type String en cliquant sur le bouton [+].
   Dans cet exemple, nommez les colonnes Id, Name, et Remove_Office.
4. Cliquez sur OK pour enregistrer vos modifications.
5. Dans le champ Number of rows, saisissez le nombre de lignes que vous souhaitez générer.
6. Dans la zone Mode, sélectionnez le mode Use Single Table.
7. Dans les champs Value, saisissez les valeurs correspondant à chaque colonne du schéma.
   Dans cet exemple, le bureau de Paris sera supprimé.

Paramètres simples du tMDMOutput

Procédure
1. Dans l’espace de modélisation graphique, cliquez sur le tMDMOutput pour ouvrir sa vue Basic Settings.
2. Dans la liste **Input Schema**, sélectionnez **Built-In** puis cliquez sur **Sync columns**.

Après avoir reçu les données du précédent composant, le composant **tMDMOutput** génère un document XML, l’écrit dans un champ de sortie et l’envoie au serveur MDM.

3. Cliquez sue **OK** pour passer à l’étape suivante.
La liste **Result of the XML serialization** dans l’onglet **Basic settings** est automatiquement renseignée avec la colonne *xml* de sortie que vous venez de créer.

4. Dans le champ **URL**, saisissez l’URL de connexion au serveur MDM.

5. Dans les champs **Username** et **Password**, saisissez l’identifiant et le mot de passe de connexion au serveur MDM.

6. Dans le champ **Data Model**, saisissez entre guillemets le nom du modèle de données par rapport auquel vous souhaitez valider les données maîtres que vous souhaitez écrire.

7. Dans le champ **Data Container**, saisissez entre guillemets le nom du conteneur de données dans lequel les données maîtres doivent être écrites.

8. Dans la zone **Partial Update**, cochez la case **Use Partial Update**.
   
   Dans le champ **Source Name** qui s’affiche avec votre sélection, saisissez le nom à utiliser dans le rapport de modifications.

   
   Dans cet exemple, saisissez “Agency/Offices/Office”.

10. Cochez la case **Delete**, puis saisissez “.” dans le champ **Key**.

**Paramètres avancés du tMDMOutput**

**Procédure**

1. Dans la vue **Component**, cliquez sur **Advanced settings** pour définir les paramètres avancés du composant *tMDMOutput*.

2. Cliquez sur le bouton [...] du champ **Configure XML Tree** pour ouvrir l’éditeur du composant. Vous pouvez également double-cliquer sur le *tMDMOutput* pour ouvrir son éditeur.

3. Dans la zone **Link target** à droite, cliquez dans le champ **Xml Tree** et remplacez **rootTag** par le nom de l’entité métier dans laquelle vous souhaitez insérer l’enregistrement de données, *Agency* dans cet exemple.
4. Dans la zone **Linker source**, sélectionnez les deux colonnes du schéma *Id* et *Name* et glissez-les sur le nœud *Agency*.
La boîte de dialogue [Selection] s'ouvre.
Sélectionnez *Create as sub-element of target node* pour que les deux colonnes correspondent aux deux sous-éléments du nœud *Agency* en XML.

5. Cliquez-droit sur le nœud racine *Agency* puis sélectionnez *Add Sub-element*.
Dans la boîte de dialogue qui s’ouvre, saisissez un nom pour les nouveaux sous-éléments, *Offices* dans cet exemple.
Répétez la même procédure pour créer un nouveau sous-élément *Office* pour le nœud *Offices* qui correspond à l’élément multi-occurrences *Offices* de l’entité métier *Agency*.

La boîte de dialogue [Selection] s’ouvre.
Sélectionnez *Create as sub-element of target node* pour relier la colonne au sous-élément du nœud *Offices* en XML.

7. Cliquez sur **Ok** pour passer à l’étape suivante.

8. Dans la zone **Link target** cliquez-droit sur l’élément que vous souhaitez définir comme élément sur lequel effectuer une boucle, puis sélectionnez *Set as Loop Element* dans le menu contextuel.
Dans cet exemple, *Id* est l’élément d’itération.

9. Cliquez sur **OK** pour valider vos changements et fermer la boîte de dialogue.

**Enregistrer et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour enregistrer votre Job.
2. Exécutez le Job en appuyant sur **F6** ou en cliquant sur **Run** dans l’onglet **Run**.
   Le bureau de l’agence situé à *Paris* avec l’identifiant d’agence *PA05* est supprimé de l’entité métier *Agency* dans le conteneur de données *DStar* du serveur MDM.
### Master Data Container Browser DStar 0.1

<table>
<thead>
<tr>
<th>Date</th>
<th>Entity</th>
<th>Keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>20160120 13:53:29</td>
<td>Agency</td>
<td>UT05</td>
</tr>
<tr>
<td>20160120 13:53:58</td>
<td>Agency</td>
<td>NV01</td>
</tr>
<tr>
<td>20160120 15:10:48</td>
<td>Agency</td>
<td>PA05</td>
</tr>
</tbody>
</table>

![XML Editor/Viewer](image)

- **Data Model**: DStar
- **Actions**: Save, Cancel

- **Options**
  - Fire a change event (update report) and triggers
  - Enable verification by before processes
tMDMReceive

Ce composant décode un paramètre de contexte comprenant des données MDM XML et les transforme en schéma plat.

Le composant tMDMReceive reçoit un enregistrement MDM provenant de déclencheurs MDM ou de processus MDM.

Propriétés du tMDMReceive Standard

Ces propriétés sont utilisées pour configurer le tMDMReceive s’exécutant dans le framework de Jobs Standard.

Le composant tMDMReceive Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier du Repository dans lequel les propriétés sont stockées. Les champs qui suivent sont automatiquement renseignés via les données récupérées.</td>
</tr>
</tbody>
</table>

Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Repository**: Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*.

**XML Record**
Saisissez le paramètre de contexte vous permettant de récupérer les dernières modifications apportées au serveur MDM. Pour plus d’informations à propos de la création et de l’utilisation des paramètres de contexte, consultez le *Guide utilisateur du Studio Talend*.

**XPath Prefix**
Si nécessaire, sélectionnez dans la liste l’expression XPath de boucle, qui est la concaténation du préfixe et du chemin de la boucle.

* /item : sélectionnez ce préfixe XPath lorsque le composant reçoit l’enregistrement d’un processus, car les processus encapsulent l’enregistrement dans un élément “item” uniquement.

* /exchange/item : sélectionnez ce préfixe XPath lorsque le composant reçoit l’enregistrement d’un déclencheur, car les déclencheurs encapsulent l’enregistrement dans un élément “item” dans un élément “exchange”.

**Loop XPath query**
Nœud de l’arborescence XML sur lequel est basée la boucle.

**Mapping**

* **Column** : reflète le schéma tel qu’il est défini dans l’éditeur Edit Schema.

* **XPath query** : Saisissez les champs à extraire de la structure XML d’entrée.

* **Get Nodes** : Cochez cette case pour ajouter les balises XML entourant les données à récupérer.

**Limit**
Nombre maximum de lignes à traiter. Si Limit = 0, aucune ligne n’est lue ou traitée.

**Die on error**
Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject.

**Advanced settings**

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

* **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

## Utilisation

| Règle d’utilisation | Ce composant est un composant de sortie. Il nécessite un flux de sortie. |

## Scénario : Extraire des informations d’un enregistrement MDM au format XML

Ce scénario s’applique uniquement aux solutions Talend avec MDM.

Le scénario suivant décrit un Job simple extrayant des informations d’un enregistrement MDM au format XML et affiche ces informations dans la console.

### Prérequis du scénario

Un conteneur de données **Product** et un modèle de données **Product** sont créés et déploïés dans le serveur MDM. Les entités de données **Product** et **Store** sont définies et des enregistrements de données existent déjà au sein de ces entités.

Les entités **Product** et **Store** sont liées par une clé étrangère, le nom du magasin (**Name** et **Store**).

Cet exemple est conçu pour obtenir des informations concernant un nouveau produit. Lorsque vous créez un enregistrement **Product**, assurez-vous que les informations **Store** sont bien ajoutées pour le nouvel enregistrement.

Les entités et leurs attributs sont affichés ci-dessous.

Pour plus d’informations concernant les principes de fonctionnement de MDM, consultez la partie MDM dans le **Guide utilisateur du Studio Talend**.
Construire le Job

Procédure
1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : un tMDMReceive et un tLogRow.
2. Reliez le tMDMReceive au tLogRow à l'aide d'un lien Row > Main.
3. Renommez les composants afin de mieux identifier leur rôle.

Configurer les composants

Définir une variable de contexte

Procédure
2. Saisissez la valeur de la variable dans le champ Value.
Notez que l’enregistrement XML doit suivre un schéma particulier. Pour plus d’informations concernant le schéma, consultez la description des processus et schémas utilisés dans les processus MDM d’appels de Jobs, dans le Guide utilisateur du Studio Talend.

Ci-dessous se trouve un échantillon de l’enregistrement XML de l’Update Report :

```xml
<exchange xmlns:mdm="java:com.amalto.core.plugin.base.xslt.MdmExtension">
  <report>
    <Update>
      <UserName>administrator</UserName>
      <Source>genericUI</Source>
      <TimeInMillis>1381486872930</TimeInMillis>
      <OperationType>ACTION</OperationType>
      <RevisionID>null</RevisionID>
      <DataCluster>Product</DataCluster>
      <DataModel>Product</DataModel>
      <Concept>Product</Concept>
      <Key>2</Key>
    </Update>
    <item><Product><Id>001</Id><Name>Computer</Name><Description>Laptop series</Description><Availability>true</Availability><Price>400</Price><OnlineStore>TalendShop@http://www.cafepress.com/Talend.2</OnlineStore><Stores><Store>[Dell]</Store><Store>[Lenovo]</Store></Stores></Product></item>
  </report>
</exchange>
```

Dans cet exemple, les espaces blancs de l’enregistrement XML sont retirés :

```xml
<exchange><report/><item><Product><Id>001</Id><Name>Computer</Name><Description>Laptop series</Description><Availability>true</Availability><Price>400</Price><OnlineStore>TalendShop@http://www.cafepress.com/Talend.2</OnlineStore><Stores><Store>[Dell]</Store><Store>[Lenovo]</Store></Stores></Product></item></exchange>
```

3. Appuyez sur les touches Ctrl+S afin de sauvegarder vos modifications.
Configurer le tMDMReceive et le tLogRow

Procédure

1. Double-cliquez sur le tMDMReceive pour ouvrir sa vue Basic settings.

2. Cliquez sur le bouton [...] à côté du champ Edit schema pour définir la structure des données. Dans cet exemple, trois colonnes sont ajoutées : Product_ID, Product_Name et Store_Name.

3. Dans le champ XML Record, saisissez la variable de contexte context.exchangeMessage.

4. Dans la liste XPath Prefix, sélectionnez "/exchange/item".

5. Dans le champ Loop XPath query, saisissez le nom de la balise racine de l’arbre XML. Dans cet exemple, saisissez "/Product/Stores/Store".

6. La colonne Column de la table Mapping est alimentée par les colonnes définies dans le schéma. Dans la colonne XPath query, saisissez votre requête XPath. Dans cet exemple, l’ID du produit, le nom du produit et le nom du magasin où se trouve le produit sont les informations extraites.

7. Double-cliquez sur le composant tLogRow pour ouvrir sa vue Basic settings.

8. Sélectionnez l’option Table (print values in cells of a table) dans la zone Mode.
Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.
2. Exécutez le Job en appuyant sur la touche **F6** ou en cliquant sur le bouton **Run** dans l’onglet **Run**.

Les informations des produits sont extraites de l’enregistrement XML et affichées dans la console.

```plaintext
[statistics] connecting to socket on port 3338
[statistics] connected
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Product_Info</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Product_ID</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>001</td>
</tr>
<tr>
<td>001</td>
</tr>
</tbody>
</table>
```

```plaintext
[statistics] disconnected

Job Product_info ended at 17:46 17/10/2013. [exit code=0]
```
tMDMRollback

Ce composant annule les modifications effectuées dans la base de données et de ne pas les commiter définitivement, notamment pour éviter les commits partiels si une erreur survient.

Le tMDMRollback rétablit l’état original de la base de données, avant l’exécution d’un Job, au lieu de commiter les modifications.

Propriétés du tMDMRollback Standard

Ces propriétés sont utilisées pour configurer le tMDMRollback s’exécutant dans le framework de Jobs Standard.

Le composant tMDMRollback Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

Basic settings

| Component List | Sélectionnez le composant tMDMConnection pour lequel effectuer l’action de rollback. |
| Close Connection | Cochez cette case pour fermer la session de la connexion au serveur MDM après avoir annulé les modifications. Notez que, même si la case n’est pas cochée, la connexion ne peut toujours pas être utilisée par le sous-job suivant, sauf si le mode Auto-Commit est activé. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau des composants. |

Global Variables


## Utilisation

| Règle d'utilisation | Ce composant est généralement utilisé avec les composants tMDMConnection, tMDMCommit, tMDMSP, tMDMViewSearch, tMDMInput, tMDMDelete, tMDMRouteRecord, tMDMOutput et tMDMClose. |

### Scénario associé

Pour un scénario associé, consultez Scénario : Supprimer des données maître d’un hub MDM à la page 2265.
tMDMRouteRecord

Ce composant permet à l’Event Manager d’identifier les modifications apportées à vos données, pour que les actions corrélatées puissent être déclenchées.

Le tMDMRouteRecord soumet la clé primaire d’un enregistrement stocké dans votre hub MDM à l’Event Manager, afin que ce gestionnaire déclenche le ou les processus attendu(s) selon certaines conditions que vous devez définir dans les pages des processus et des déclencheurs de la perspective MDM du Studio Talend.

Pour plus d’informations concernant l’Event Manager, consultez le Guide utilisateur du Studio Talend.

Propriétés du tMDMRouteRecord Standard

Ces propriétés sont utilisées pour configurer le tMDMRouteRecord s’exécutant dans le framework de Jobs Standard.

Le composant tMDMRouteRecord Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case pour utiliser un composant tMDMConnection déjà configuré.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDM version</td>
<td>Par défaut, l’option Server 6.0 est sélectionnée. Bien qu’il soit recommandé de migrer les Jobs existants vers cette nouvelle version, l’option Server 5.6 est disponible pour faciliter le processus de migration de vos Jobs, afin qu’ils continuent à fonctionner avec un serveur 6.0 sans modification. Pour ce faire, une option sur le serveur doit être activée pour accepter et traduire les requêtes de ces Jobs.</td>
</tr>
<tr>
<td>URL</td>
<td>Saisissez l’URL d’accès au serveur MDM.</td>
</tr>
</tbody>
</table>
| Username et Password      | Informations d’authentification de l’utilisateur du serveur MDM.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
<p>| Data Container            | Saisissez le nom du conteneur de données contenant les données que vous souhaitez effacer. |
| Type                      | Sélectionnez Master ou Staging pour spécifier sur quelle base de données effectuer l’action. |
| Entity Name               | Saisissez le nom de l’entité métier contenant l’enregistrement que vous souhaitez faire lire à l’Event Manager. |</p>
<table>
<thead>
<tr>
<th><strong>IDs</strong></th>
<th>Spécifiez la (les) clé(s) primaire(s) de(s) l’enregistrement(s) que vous souhaitez faire lire à l’<strong>Event Manager</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advanced settings</strong></td>
<td></td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
<tr>
<td><strong>Global Variables</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Global Variables</strong></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option. Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
<tr>
<td><strong>Utilisation</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Règle d’utilisation</strong></td>
<td>Ce composant est un composant de début. Il nécessite un flux de sortie. Vous pouvez augmenter le délai avant suspension de la connexion pour un Job utilisant ce composant afin de faciliter le traitement d’un grand volume d’enregistrements de données. Pour plus d’informations, consultez les paramètres d’exécution avancés pour les paramètres JVM sur <strong>Talend Help Center</strong> (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>
| **Connections** | Liens de sortie (de ce composant à un autre) :  
  **Row** : Iterate.  
Liens d’entrée (d’un autre composant à celui-ci) :  
  **Row** : Iterate.  
Pour plus d’informations concernant les connexions, consultez la section relative aux types de connexion, dans le **Guide utilisateur du Studio Talend**. |
Scénario : Router un rapport de mise à jour vers l'Event Manager

Ce scénario s’applique uniquement aux solutions Talend avec MDM.
Dans ce scénario, le composant tMDMRouteRecord est utilisé pour soumettre à l’Event Manager la clé primaire d’un rapport de mise à jour à l’Event Manager, afin de lancer un déclencheur appelant un Job qui affiche la notification indiquant qu’un enregistrement de données a été mis à jour.

Prérequis

- Le serveur MDM doit être en cours de fonctionnement.
- Le projet démo MDM doit avoir été importé.
- Le conteneur de données Product doit avoir été alimenté avec des enregistrements de données.

Pour plus d’informations concernant la création d’un conteneur de données et d’un modèle de données, consultez le Guide utilisateur du Studio Talend.
Pour plus d’informations concernant la création et l’accès aux attributs visualisables, consultez le Guide utilisateur de Talend MDM Web UI.

Créer un Job notifiant les utilisateurs qu’un enregistrement de données a été mis à jour

Dans ce scénario, un Job nommé message est créé avec un seul composant tJava informant les utilisateurs de la mise à jour d’un enregistrement de données existant.

Procédure

1. Déposez un composant tJava dans l’espace de modélisation graphique.
2. Double-cliquez sur le composant pour afficher sa vue Basic settings.
3. Dans le champ Code, saisissez `System.out.println("------- A Product is Updated.---" + new java.util.Date());`.
4. Sauvegardez votre Job et déployez-le sur le serveur MDM.

Résultats

Pour plus d’informations concernant le composant tJava, consultez tJava à la page 1939.

Pour plus d’informations concernant le déploiement d’un Job sur le serveur MDM, consultez le Guide utilisateur du Studio Talend.

Générer un Job appelant un déclencheur et le déployant sur le serveur MDM

Dans ce scénario, une fois le Job message créé, un déclencheur appelant ce Job peut être généré.

Procédure

1. Dans la vue MDM Repository, sous le nœud Job Designs, cliquez-droit sur le Job message.
2. Dans le menu contextuel, sélectionnez Generate Talend Job Caller Trigger, acceptez les options par défaut et cliquez sur Generate.
Un déclencheur utilisé pour appeler le Job message Job est généré et stocké sous Event Management > Trigger. Son nom est CallJob_Message et la valeur de son champ Service JNDI Name est callJob.

3. Dans la zone Trigger XPath Expressions, personnalisez les conditions du déclencheur selon vos besoins.
Dans cet exemple, ajoutez les conditions suivantes :

<table>
<thead>
<tr>
<th>XPath</th>
<th>Opérateur</th>
<th>Valeur</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update/DataCluster</td>
<td>Matches</td>
<td>Product</td>
<td>C1</td>
</tr>
<tr>
<td>Update/DataModel</td>
<td>Matches</td>
<td>Product</td>
<td>C2</td>
</tr>
<tr>
<td>Update/OperationType</td>
<td>Matches</td>
<td>UPDATE</td>
<td>C3</td>
</tr>
</tbody>
</table>

4. Dans la zone Conditions, combinez les conditions à l’aide des prédicats.
Dans cet exemple, saisissez :

C1 AND C2 AND C3

Ainsi, ce déclencheur sera exécuté sur un enregistrement dans l’entité Update, uniquement lorsqu’un enregistrement de données Product est mis à jour.

5. Déployez le déclencheur personnalisé sur le serveur MDM.

Mettre à jour d’un enregistrement de données dans un conteneur de données spécifique

Procédure
1. Connectez-vous à Talend MDM Web UI.
   Pour plus d’informations concernant la connexion à Talend MDM Web UI, consultez le Guide utilisateur de Talend MDM Web UI.
2. Dans le coin supérieur droit de la page Web, vérifiez que le panneau Actions est affiché.
3. Dans la zone Domain Configuration, sélectionnez le conteneur et le modèle de données requis.
   Dans cet exemple, sélectionnez Product pour le conteneur et le modèle de données.
4. Cliquez sur Save afin de sauvegarder vos modifications.
5. Allez à la page Master Data Browser et sélectionnez l’entité Product dans la liste.
6. Double-cliquez sur l’un des enregistrements de données pour afficher ses attributs visualisables dans une nouvelle vue dédié à ce produit.
   Dans cet exemple, ouvrez l’enregistrement du produit Talend Mug d’ID unique 231035938.
7. Mettez à jour une des valeurs d’attribut dans l’enregistrement du produit.
   Dans cet exemple, mettez à jour l’enregistrement en cochant la case Availability pour le rendre disponible.
8. Cliquez sur Save afin de valider cette mise à jour.
**Accéder aux rapports Update Reports**

**Procédure**

1. Dans la vue **MDM Repository**, double-cliquez sur **Data Container > System > UpdateReport** pour ouvrir la vue **UpdateReport**. **UpdateReport** contient les traces complètes de toutes les actions de création, de mise à jour ou de suppression (Create, Update ou Delete) effectuées sur les données maître.

   Le rapport de mise à jour (Update Report) contient les traces complètes de toutes les actions de création, de mise à jour ou de suppression (create, update ou delete) effectuées sur les données maître.

   Si le conteneur de données UpdateReport n'est pas disponible, vous devez d'abord l’importer de votre serveur MDM. Pour plus d'informations concernant l'import d’objets système depuis le serveur MDM dans votre référentiel local, consultez le Guide utilisateur du Studio Talend.

2. À côté du champ **Entity** de cette vue, cliquez sur le bouton ![Recherche](image) pour rechercher tous les enregistrements correspondant à votre action de mise à jour de l’enregistrement de données comme décrit dans Mettre à jour d’un enregistrement de données dans un conteneur de données spécifique à la page 2315.

   Notez que l’entité **Update** ne signifie pas nécessairement que l’action enregistrée est une mise à jour, c’est simplement le nom de l’entité définie par le modèle de données du rapport **UpdateReport** pouvant contenir différentes actions comme create, delete et update.

3. Trouvez l'emplacement de l'enregistrement correspondant à votre action de mise à jour de l’enregistrement de données comme décrit dans Mettre à jour d’un enregistrement de données dans un conteneur de données spécifique à la page 2315.

   L’enregistrement du rapport de mise à jour sera routé vers l’**Event Manager**, dont la clé primaire est **genericUI.1499236505686**.

**Construire le Job pour router l’enregistrement du rapport de mise à jour à l’Event Manager**

**Procédure**

1. Dans la perspective Integration, cliquez-droit sur **Job Designs**, dans la vue **Repository** et, dans le menu contextuel, sélectionnez **Create Standard Job**.

2. Saisissez **RouteRecord** dans le champ **Name** de l’assistant qui s’ouvre, puis cliquez sur **Finish**.

3. Déposez le composant **tMDMRouteRecord** de la Palette dans l’espace de modélisation graphique.

4. Double-cliquez sur ce composant afin d’ouvrir sa vue **Component**.

5. Dans le champ **URL**, saisissez l’adresse d’accès à votre serveur MDM.

   Dans cet exemple, saisissez **http://localhost:8180/talendmdm/services/soap**.

6. Dans les champs **Username** et **Password**, saisissez les identifiants d'accès au serveur MDM.

7. Dans le champ **Data Container** saisissez le nom du conteneur de données contenant l’enregistrement que vous souhaitez router vers l’**Event Manager**.

   Dans cet exemple, saisissez **UpdateReport**.

8. Dans le champ **Entity Name**, saisissez le nom de l’entité à laquelle appartient l’enregistrement que vous souhaitez router.

   Dans cet exemple, son nom est **Update**.

9. Dans la zone **IDS**, cliquez sur le bouton 

   Dans le champ **Entity Name**, saisissez le nom de l’entité à laquelle appartient l’enregistrement que vous souhaitez router.
Dans cet exemple, la clé primaire est `genericUI.1499236505686`.

Sauvegarder et exécuter le Job

Procédure
1. Appuyez sur les touches Ctrl+S afin de sauvegarder le Job.

Dans cet exemple, puisque l’enregistrement de données Product d’ID unique `231035938` est mis à jour, le composant tMDMRouteRecord soumet la clé primaire de l’enregistrement du rapport de mise à jour correspondant à l’Event Manager. Lorsque l’Event Manager vérifie cet enregistrement et confirme que cet enregistrement répond bien aux conditions dans la vue de configuration du déclencheur CallJob_Message, l’Event Manager appelle le déclencheur pour appeler le Job message.

3. Vérifiez les résultats d’exécution du Job en consultant le log du serveur MDM pour voir que la notification `------- A Product is Updated.-------` est affichée.

Vous pouvez vérifier que le serveur enregistre d’une de ces deux manières :

- Dans le panneau Server Explorer, cliquez-droit sur la connexion au serveur où est installé le serveur MDM, puis cliquez sur View Server Log. Une console s’ouvre, affichant le contenu le plus récent du fichier mdm.log.

- Vous pouvez également voir le contenu du fichier mdm.log dans un navigateur. Pour plus d’informations, consultez le Guide utilisateur du Studio Talend.

- Si vous avez un accès direct au serveur MDM, vous pouvez également consulter le fichier directement dans le répertoire `<$INSTALLDIR>/log`, où INSTALLDIR indique l’emplacement où est installé le serveur MDM.
tMDMSP

Ce composant centralise des requêtes multiples ou complexes dans un hub MDM et appelle une procédure stockée.

**Propriétés du tMDMSP Standard**

Ces propriétés sont utilisées pour configurer le tMDMSP s’exécutant dans le framework de Jobs Standard.

Le composant tMDMSP Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

**Basic settings**

| **Schema et Edit Schema** | Dans une procédure stockée, le schéma est un paramètre d’entrée.  
Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant.  
Le schéma de ce composant est en lecture seule. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant tMDMConnection adéquat à partir de la liste Component pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>MDM version</strong></td>
<td>Par défaut, l’option Server 6.0 est sélectionnée. Même s’il est recommandé de migrer des Jobs existants vers cette nouvelle version, l’option Server 5.6 est disponible afin de simplifier le processus de migration de vos Jobs, pour qu’ils continuent à fonctionner sans modification, avec un serveur 6.0. Pour ce faire, une option sur le serveur doit être activée pour accepter et traduire les requêtes de ces Jobs.</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td>Saisissez l’URL du serveur MDM.</td>
</tr>
</tbody>
</table>
| **Username et Password** | Informations d’authentification sur l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Data Container** | Saisissez le nom du conteneur de données contenant les données que vous souhaitez effacer. |
| **Type** | Sélectionnez Master ou Staging pour spécifier sur quelle base de données effectuer l’action. |
### Procedure Name

Saisissez le nom exact de la Procédure Stockée.

### Parameters (in order)

Cliquez sur le bouton 
[+] et sélectionnez les colonnes d'entrée (Input Columns) requises pour les procédures.

**Remarque :**

Le schéma SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure.

### Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.

### Global Variables

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

**Règle d’utilisation**

Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début mais aucun paramètre d’entrée n’est requis pour la procédure à appeler. Un lien de sortie est requis.

Vous pouvez augmenter le délai avant suspension de la connexion pour un Job utilisant ce composant afin de faciliter le traitement d’un grand volume d’enregistrements de données. Pour plus d’informations, consultez les paramètres d’exécution avancés pour les paramètres JVM sur **Talend Help Center** (https://help.talend.com).

**Connections**

- **Row** : Main


Liens d’entrée (d’un composant à l’autre) :

Liens de sortie (d’un composant à l’autre) :
Scénario : Exécuter une procédure stockée à l’aide du tMDMSP

Ce scénario s’applique uniquement aux solutions Talend avec MDM.
Dans ce scénario, le Job génère d’abord des paramètres et les envoie au tMDMSP, qui exécute une procédure stockée prédéfinie, puis extrait les données du résultat d’exécution retourné et les affiche dans la console.

Prérequis :
• Vérifiez que le serveur MDM est en cours de fonctionnement.
• Vous avez importé le projet démo MDM et chargé les données d’exemple dans le conteneur de données Product, en exécutant le Job MDM_LoadAll.
• Vous avez créé une procédure stockée nommée ProductSelection et déployé cette procédure stockée sur le serveur MDM.

Dans cet exemple, la procédure stockée ProductSelection est conçue pour interroger deux champs Name et Price des enregistrements de données Product, dans un intervalle de prix donné :

Pour plus d’informations concernant l’utilisation de procédures stockées, consultez le Guide utilisateur du Studio Talend.
Créer un Job exécutant la procédure stockée

Procédure
1. Glissez-déposez les composants suivants dans l’espace de modélisation graphique : un **tFixedFlowInput**, **tMDMSP**, un **tExtractXMLField** et un **tLogRow**.
2. Reliez les composants à l’aide de liens **Row > Main**.

Configurer le tFixedFlowInput pour générer un intervalle de prix

Procédure
1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

![Diagramme de tFixedFlowInput](image)

3. Cliquez deux fois sur le bouton [+], pour ajouter deux paramètres **min** et **max**, utilisés pour définir l’intervalle de prix.

![Diagramme de gestion des paramètres min et max](image)

5. Dans la zone **Mode**, laissez sélectionnée l’option par défaut **Use Single Table** et saisissez 10, entre guillemets, pour le paramètre **min** et 15, entre guillemets, pour le paramètre **max**.

**Configurer le tMDMSP pour exécuter la procédure stockée**

**Procédure**

1. Double-cliquez sur le tMDMSP pour ouvrir sa vue **Basic settings**.

2. Saisissez l’identifiant et le mot de passe pour accéder au serveur MDM.


5. Dans la table **Parameters**, cliquez deux fois sur le bouton [*+] pour ajouter deux lignes et sélectionnez respectivement **min** et **max**.

**Configurer le tExtractXMLField pour extraire des données depuis les résultats d’exécution de la procédure stockée**

**Procédure**

1. Double-cliquez sur le tExtractXMLField pour ouvrir sa vue **Basic settings**.
2. Cliquez sur le bouton [ ] à côté du champ **Edit schema** pour ouvrir l’éditeur du schéma.

3. Ajoutez deux colonnes pour définir la structure des données de sortie : *name* et *price*.

4. Cliquez sur **OK** pour fermer l’éditeur du schéma et cliquez sur **Yes** dans la boîte de dialogue [Propagate].

5. Dans le champ **Loop XPath query**, saisissez *result* entre guillemets, pour baser la boucle sur cet élément.

   Chaque résultat de la procédure stockée contient la syntaxe suivante : `<result><col0>val1</col0>...<colN>valn</colN></result>` (où *colN* est la *n*-ième colonne de la clause **SELECT** clause).

   **Avertissement :**

   Si une valeur null existe dans un enregistrement de données à retourner par les résultats d’exécution de la procédure stockée, des incohérences se produiront entre le numéro de colonne et la valeur de cette colonne dans l’enregistrement de données retourné.

6. Dans la colonne **XPath query** de la table **Mapping**, saisissez le nom exact du nœud sur lequel appliquer la boucle : *col0* et *col1*, respectivement, entre guillemets.

**Configurer le mode d’affichage des données et exécuter le Job**

**Procédure**

1. Double-cliquez sur le composant **tLogRow** pour afficher sa vue **Basic settings**.

2. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour une lisibilité optimale des résultats.
3. Sauvegardez le Job et appuyez sur **F6** pour l’exécuter.

Selon la procédure stockée et l’intervalle de prix, les enregistrements de données *Product* contenus dans l’intervalle de prix de 10 à 15 sont affichés, avec les valeurs des champs *Name* et *Price*, comme attendu.

```
<table>
<thead>
<tr>
<th>tLogRow_1</th>
<th>name</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Talend Trucker Hat</td>
<td>10.99</td>
</tr>
<tr>
<td></td>
<td>Talend Mug</td>
<td>10.99</td>
</tr>
<tr>
<td></td>
<td>Talend Large Mug</td>
<td>11.99</td>
</tr>
<tr>
<td></td>
<td>Talend Stein</td>
<td>13.99</td>
</tr>
<tr>
<td></td>
<td>Talend Cap</td>
<td>14.99</td>
</tr>
</tbody>
</table>
```
tMDMTriggerInput

Ce composant lit un message XML (de type Document) envoyé par MDM et passe les informations au composant suivant.

À chaque sauvegarde d'une modification dans votre MDM, l'enregistrement modifié correspondant est généré au format XML. Lors de l'exécution, ce composant lit l'enregistrement et envoie les informations relatives au composant suivant. Avec le tMDMTriggerInput, il n'est pas nécessaire de configurer votre Job afin de le faire communiquer les modifications de données de MDM vers votre Job.

Remarque :

Ce composant fonctionne avec le nouveau service de déclenchement et le plug-in de traitement en version 5.0 et supérieure de MDM. Les Jobs, les déclencheurs et les processus MDM développés dans les versions précédentes de MDM restent supportés. Il est cependant recommandé d'utiliser ce composant lors de la création de nouveau Jobs MDM.

Propriétés du tMDMTriggerInput Standard

Ces propriétés sont utilisées pour configurer le tMDMTriggerInput s'exécutant dans le framework de Jobs Standard.

Le composant tMDMTriggerInput Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong>:</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Le tMDMTriggerInput doit utiliser cette option pour appliquer le schéma en lecture seule par défaut. La colonne MDM_message est l'unique colonne de ce schéma.</td>
</tr>
<tr>
<td><strong>Repository</strong>:</td>
<td>Sélectionnez le fichier du Repository dans lequel les propriétés sont stockées. Les champs qui suivent sont automatiquement renseignés via les données récupérées.</td>
</tr>
<tr>
<td></td>
<td>Comme le composant tMDMTriggerInput fournit un schéma fixe en lecture seule, vous devez utiliser l'option Built-in.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
</tbody>
</table>
- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-in** : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*. C'est l'option par défaut pour le **tMDMTriggerInput**. Avec cette option, le schéma en lecture seule est utilisé pour traiter le schéma du message MDM au format XML.

**Repository** : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Jobs. Voir également le *Guide utilisateur du Studio Talend*. Comme le composant **tMDMTriggerInput** fournit un schéma fixe en lecture seule, vous devez utiliser l'option **Built-in**.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. 
Utilisation

Règle d’utilisation

Ce composant est un composant de début. Il requiert un flux de sortie.

Pour recevoir le message de MDM, vous devez déployer le Job à l’aide de ce composant sur votre serveur MDM et générer le déclencheur et le processus correspondants dans MDM pour invoquer ce Job.

Pour plus d’informations concernant le déploiement d’un Job sur le serveur MDM et la génération d’un déclencheur ou d’un processus, consultez le Guide utilisateur du Studio Talend.

Pour plus d’informations concernant la modification d’un enregistrement dans MDM, consultez le Guide utilisateur de Talend MDM Web UI.

Scénario : Échanger des informations concernant les événements d’un enregistrement MDM

Ce scénario s’applique uniquement aux solutions Talend avec MDM.

Dans ce scénario, un Job à quatre composants est utilisé pour échanger des informations sur des événements concernant un enregistrement produit. À l’aide d’une connexion MDM établie depuis le Repository, ce Job est appelé par un déclencheur, une fois que vous avez mis à jour un enregistrement produit existant.

Pour reproduire ce scénario, accomplissez les tâches suivantes, dans l’ordre :

1. Créez une connexion MDM de type Receive dans le Repository du Studio. Cette connexion au hub MDM contient les enregistrements que vous souhaitez mettre à jour.

2. Créez le Job pour recevoir et envoyer le message MDM de mise à jour.

3. Générez le déclencheur invoquant ce Job.

4. Mettez à jour un enregistrement produit.

Pour créer les enregistrements, les modèles et les conteneurs de données utilisés dans ce scénario, vous pouvez exécuter les Jobs du projet démo MDM dans le Studio Talend puis mettre à jour le serveur MDM afin de déployer les objets créés et qu’ils soient pris en compte lors de l’exécution. Vous allez utiliser ce serveur à travers tout le scénario.

Vous pouvez obtenir plus d’informations concernant la gestion des événements MDM, concernant l’import d’un projet démo et concernant le déploiement d’objets sur le serveur MDM sur Talend Help Center (https://help.talend.com).

Créer une connexion MDM pour récupérer des entités

Établir la connexion

Avant de commencer

Assurez-vous que le serveur MDM auquel vous devez communiquer le message de mise à jour est en cours de fonctionnement.
**Procédure**

1. Dans la perspective **Integration** du Studio Talend, développez le nœud **Metadata** dans le **Repository**.
2. Cliquez-droit sur l’élément **Talend MDM** et sélectionnez **Create MDM connection**.

   ![MDM Connection Window](image)

3. Dans le champ **Name**, saisissez le nom à utiliser pour cette connexion et, si nécessaire, ajoutez un objectif (**Purpose**) et une **Description** dans les champs correspondants.
   Par exemple, nommez cette connexion **Receive**.

   Lorsque vous avez cliqué sur le bouton **Check** et que la connexion est établie, le bouton **Next** devient cliquable.
5. Cliquez sur **Next**, puis sélectionnez le modèle de données (**Data model**) et le conteneur de données (**Data Container**) utilisés par l'enregistrement à mettre à jour.
   Dans ce scénario, le modèle et le conteneur sont nommés **Product**.
6. Cliquez sur **Finish** afin de valider la création.
   La connexion créée est affichée sous le nœud **Metadata** dans le **Repository**.

**Récupérer des entités**

**Procédure**

1. Cliquez-droit sur la connexion créée et, dans le menu contextuel, sélectionnez **Retrieve Entity**.
2. Sélectionnez **Receive MDM** dans la boîte de dialogue et cliquez sur **Next** pour ccontinuer.
3. Sélectionnez l’entité à récupérer et cliquez sur Next.
Dans ce scénario, sélectionnez Product. Le champ Name est automatiquement renseigné.

Dans ce scénario, l’élément Features est le champ sur lequel effectuer une boucle et les éléments Id, Name et Description sont les champs à extraire.

Vous pouvez obtenir plus d’informations concernant les conteneurs de données et les modèles de données utilisés par la solution MDM de Talend sur Talend Help Center (https://help.talend.com).

**Créer le Job communiquant le message MDM**

**Relier les composants**

**Procédure**

1. Créez un Job nommé `update_product`.
2. Déposez un `tMDMTriggerInput`, un `tXMLMap`, un `tMDMTriggerOutput` et un `tLogRow` de la Palette dans l’espace de modélisation graphique.
3. Cliquez-droit sur le **tMDMTriggerInput**, dans le menu contextuel, sélectionnez le lien **Row > Main** et reliez-le au **tXMLMap**.

4. Répétez l’opération pour relier le **tXMLMap** au **tMDMTriggerOutput** et nommez le lien **output**.

5. Double-cliquez sur le **tMDMTriggerOutput** pour ouvrir sa vue **Component**.

6. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur de schéma.

7. Sélectionnez la colonne prédéfinie du **tMDMTriggerOutput**, puis cliquez sur l’icône ![reproducer](image) pour reproduire cette colonne dans le schéma d’entrée (gauche).

8. Reliez le **tMDMTriggerOutput** au **tLogRow** à l’aide d’un lien **Row > Main**.

### Configurer la transformation du message MDM

#### Procédure

1. Double-cliquez sur le **tXMLMap** pour ouvrir son éditeur.

2. Dans la table représentant le flux d’entrée (en haut à gauche de l’éditeur), cliquez-droit sur le nom de la colonne **MDM_Message**, en haut de l’arborescence XML et sélectionnez **Import From Repository**.

   L’assistant **Metadata** s’ouvre.


   Cela construit un document d’entrée complet pour un événement MDM. Dans l’arborescence XML d’entrée, l’élément **Features** est automatiquement configuré comme élément de boucle.
5. Dans la table représentant le flux de sortie (en haut à droite de l’éditeur), développez l’arborescence XML de sortie comme dans l’image ci-dessous.

Cette arborescence est construite selon le modèle statique requis du rapport de sortie MDM. La construction XML requise pour retourner le message de validation-réussite est `<report><message type="info">message</message></report>`. La construction XML requise pour retourner le message de validation-échec est `<report><message type="error">message</message></report>`. 
6. Mappez l'élément OperationType d'entrée à l'élément message de sortie. Cela va écrire en sortie les informations concernant le type de l'événement se produisant sur l'enregistrement MDM.

7. Dans la colonne Expression, saisissez "info" dans la ligne correspondant à @type.

8. Cliquez sur l'icône de clé anglaise pour afficher le panneau des paramètres de sortie et configurez l'option All in one à true.

9. Cliquez sur OK pour fermer l'éditeur et valider les modifications.

10. Double-cliquez sur le tLogRow pour ouvrir sa vue Component, puis cliquez sur Sync columns.

**Généraliser le déclencheur pour invoquer le Job créé**

**Déployer le Job à déployer sur le serveur MDM**

**Procédure**

1. Passez en perspective MDM et développez le nœud Job Designs dans la vue MDM Repository.
2. Cliquez-droit sur le Job update_product précédemment créé dans ce scénario et sélectionnez Deploy To... dans le menu contextuel. L'assistant au déploiement s'ouvre.
3. Dans la liste des serveurs, sélectionnez le serveur MDM que vous utilisez, puis cliquez sur OK.
4. Dans la fenêtre Deploy to Talend MDM qui s'ouvre, conservez les paramètres par défaut : Export type à Hosted (zip) et Context scripts à Default.
   Vous pouvez obtenir plus d’informations concernant ces paramètres sur Talend Help Center (https://help.talend.com).
5. Cliquez sur Finish afin de valider ces paramètres et démarrer le déploiement.
   Une fois le déploiement terminé, un message s’affiche pour vous indiquer la réussite du déploiement.
6. Cliquez sur OK pour fermer ce message.
   Une fenêtre s’ouvre et vous indique que le Job update_product a bien été déployé.
7. Cliquez sur OK pour finaliser le déploiement.
Générer le déclencheur utilisé pour appeler le Job

**Procédure**

2. Dans la fenêtre qui s'ouvre, conservez les paramètres par défaut pour ce scénario : `Integrated` et `Embedded`.
3. Cliquez sur `Generate` pour démarrer la génération.
   Cela fait, un déclencheur nommé `CallJob_update_product` s'affiche sous le nœud `Trigger` dans le `MDM Repository`.
4. Cliquez-droit sur ce déclencheur et sélectionnez `Deploy To...` dans le menu contextuel afin de le déployer sur le serveur MDM.
5. Dans l'assistant qui s'ouvre, sélectionnez le serveur que vous utilisez et cliquez sur `OK`.
6. Cliquez sur `OK` pour finaliser le déploiement.

Mettre à jour un enregistrement produit

**Procédure**

2. Dans le panneau `Actions`, à droite, sélectionnez pour `Data Container` et `Data Model`.
3. Dans la page `Master Data Browser`, lancez la recherche dans l'entité `Product` pour lister tous les enregistrements produits disponibles.
4. Sélectionnez dans la liste l'enregistrement produit à mettre à jour, par exemple, `Talend Trucker Hat`.
   Les détails de cet enregistrement sont affichés dans la vue `Product`.
5. Mettez à jour l'un des attributs.
   Dans cet exemple, mettez à jour le prix pour qu'il devienne `11.00` et cliquez sur `Save`.
   Le message concernant le type d'opération de cet événement a été envoyé au serveur MDM. Ce message est affiché dans la fenêtre du serveur MDM.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<root><report><message_type="info">UPDATE</message><message></message></report></root>
```
**tMDMTriggerOutput**

Ce composant reçoit un flux XML (de type `Document`) du composant précédent dans le Job.

Le `tMDMTriggerOutput` reçoit un flux XML afin de configurer un message MDM pour que MDM récupère le message lors de l'exécution. Avec ce composant, il n’est pas nécessaire de configurer votre Job afin de le faire communiquer les modifications de données de MDM vers votre Job.

**Remarque :**

Ce composant fonctionne avec le nouveau service de déclenchement et le plug-in de traitement en version 5.0 et supérieure de MDM. Les Jobs, les déclencheurs et les processus MDM développés dans les versions précédentes de MDM restent supportés. Il est cependant recommandé d’utiliser ce composant lors de la création de nouveau Jobs MDM.

**Propriétés du tMDMTriggerOutput Standard**

Ces propriétés sont utilisées pour configurer le `tMDMTriggerOutput` s’exécutant dans le framework de Jobs Standard.

Le composant `tMDMTriggerOutput Standard` appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Le composant <code>tMDMTriggerOutput</code> doit utiliser cette option pour appliquer le schéma en lecture seule par défaut. La colonne <code>MDM_message</code> est l’unique colonne de ce schéma.</td>
<td></td>
</tr>
</tbody>
</table>

| **Repository** | Sélectionnez le fichier du Repository dans lequel les propriétés sont stockées. Les champs qui suivent sont automatiquement renseignés via les données récupérées. |
| Comme le `tMDMTriggerOutput` fournit un schéma fixe en lecture seule, vous devez utiliser l’option Built-in. |

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.

Cliquez sur `Edit schema` pour modifier le schéma. Si le schéma est en mode `Repository`, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
**Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Built-in** : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

C'est l'option par défaut pour le **tMDMTriggerOutput**. Avec cette option, le schéma en lecture seule est utilisé pour traiter le schéma du message MDM au format XML.


Comme le composant **tMDMTriggerOutput** fournit un schéma fixe en lecture seule, vous devez utiliser l’option **Built-in**.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

**Règle d’utilisation**

Ce composant est un composant de fin. Il requiert un flux d’entrée.

Pour envoyer un message au MDM, vous devez déployer le Job à l’aide de ce composant sur votre serveur MDM et
générer le déclencheur et le processus correspondants afin d’invoquer ce Job dans MDM.

Pour plus d’informations concernant le déploiement d’un Job sur le serveur MDM et la génération d’un déclencheur ou d’un processus, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Durant le déploiement de ce composant sur le serveur MDM, vous devez sélectionner le type <strong>Hosted (Zip)</strong> comme format du Job déployé. Si vous le déployez dans le type <strong>Distributed (War)</strong>, le Job relatif ne peut être invoqué. Pour plus d’informations concernant les types disponibles, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez Scénario : Échanger des informations concernant les événements d’un enregistrement MDM à la page 2327.
tMDMViewSearch

Ce composant récupère des enregistrements MDM d'un hub MDM en appliquant les filtres que vous avez créés dans une Vue spécifique et écrit les résultats dans une structure XML.

Pour plus d’informations concernant la vue permettant de définir des critères de filtres, consultez le Guide utilisateur du Studio Talend.

Propriétés du tMDMViewSearch Standard

Ces propriétés sont utilisées pour configurer le tMDMViewSearch s'exécutant dans le framework de Jobs Standard.

Le composant tMDMViewSearch Standard appartient à la famille Talend MDM.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma soit local (built-in) soit distant dans le Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Sync columns afin de récupérer le schéma du composant précédent.</td>
</tr>
</tbody>
</table>

|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| XML Field | Sélectionnez le nom de la colonne dans laquelle vous souhaitez écrire les données XML. |

2339
<table>
<thead>
<tr>
<th><strong>Use an existing connection</strong></th>
<th>Cochez cette case et sélectionnez le composant tMDMConnection adéquat à partir de la liste Component pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MDM version</strong></td>
<td>Par défaut, l’option Server 6.0 est sélectionnée. Bien qu’il soit recommandé de migrer les Jobs existants vers cette nouvelle version, l’option Server 5.6 est disponible pour faciliter le processus de migration de vos Jobs, afin qu’ils continuent à fonctionner avec un serveur 6.0 sans modification. Pour ce faire, une option sur le serveur doit être activée pour accepter et traduire les requêtes de ces Jobs.</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td>Saisissez l’URL du serveur MDM.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification sur l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Data Container</strong></td>
<td>Saisissez le nom du conteneur de données contenant les données que vous souhaitez effacer.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Sélectionnez Master ou Staging pour spécifier sur quelle base de données effectuer l’action.</td>
</tr>
<tr>
<td><strong>View Name</strong></td>
<td>Saisissez le nom de la vue dont les filtres seront appliqués pour traiter les enregistrements.</td>
</tr>
<tr>
<td><strong>Operations</strong></td>
<td>Remplissez ce tableau afin de créer une clause WHERE. Les paramètres à définir sont :&lt;br&gt;  - <strong>XPath</strong> : définissez le chemin de l’expression afin de sélectionner le nœud XML sur lequel appliquer les filtres.&lt;br&gt;  - <strong>Functions</strong> : sélectionnez un opérateur dans la liste déroulante, comme Contains, Starts with, Equals, etc.&lt;br&gt;  - <strong>Value</strong> : saisissez la valeur que vous souhaitez récupérer. La valeur peut être une chaîne de caractères (par exemple &quot;Apple&quot;, un élément Entité (par exemple &quot;Product/Name&quot;) ou une variable personnalisée (par exemple &quot;$user_context.language&quot; ou &quot;$user_context/properties['location']&quot;).&lt;br&gt;  - <strong>Predicate</strong> : sélectionnez le prédicat permettant de combiner les conditions de filtres de différentes manières. Le prédicat peut être none, or, and, exactly, etc.</td>
</tr>
<tr>
<td><strong>Avertissement</strong></td>
<td>Les paramètres sont sensibles à la casse.</td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant les opérateurs et les prédicats, consultez Propriétés du tMDMInput Standard à la page 2272.

| Order (One Row) | Remplissez ce tableau afin de choisir l’ordre de présentation des enregistrements récupérés. Les paramètres à définir sont :
- **XPath** : définissez le chemin de l’expression afin de sélectionner le nœud XML sur lequel appliquer le tri.
- **Order** : sélectionnez l’ordre de présentation, soit **asc** (ascendant) ou **desc** (descendant).

**Avertissement** :
Les paramètres sont sensibles à la casse.

**Avertissement** :
Pour le moment, seule la première ligne créée dans le tableau **Order** est valide.

| Spell Threshold | Configurez cette option sur -1 afin de désactiver le seuil. Ce seuil est utilisé pour décider du niveau de tolérance des erreurs orthographiques.

| Skip Rows | Renseignez le nombre de lignes à ignorer afin de spécifier à partir de quelle ligne doit commencer le processus. Par exemple, si vous saisissez 8, le processus commencera à la neuvième ligne.

| Max Rows | Saisissez le nombre maximal de lignes à traiter. Si le champ **Limit** est paramétré sur 0, aucune ligne ne sera lue ou traitée. Par défaut, la valeur de ce champ est -1, ce qui signifie qu’aucune limite n’est définie.

**Advanced settings**

| Batch Size | Nombre de lignes dans chaque lot traité. Par défaut, la valeur de ce champ est -1, ce qui signifie que toutes les lignes sont traitées dans un seul lot.

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

**Global Variables**

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaine de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l'aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

**Règle d’utilisation**

Utilisez ce composant pour récupérer des enregistrements spécifiques.

Vous pouvez augmenter le délai avant suspension de la connexion pour un Job utilisant ce composant afin de faciliter le traitement d’un grand volume d’enregistrements de données. Pour plus d’informations, consultez les paramètres d’exécution avancés pour les paramètres JVM sur **Talend Help Center** (https://help.talend.com).

**Connections**

Liens de sortie (de ce composant à un autre) :

- **Row** : Iterate

Liens d’entrée (d’un autre composant à celui-ci) :

- **Row** : Iterate;

Pour plus d’informations concernant les connexions, consultez la section relative aux types de connexion, dans le **Guide utilisateur du Studio Talend**.

### Scénario : Récupérer des enregistrements d’un hub MDM via une vue existante

Ce scénario s’applique uniquement aux solutions Talend avec MDM.

Ce scénario décrit un Job comprenant deux composants récupérant un enregistrement de données ayant une structure XML.

Dans cet exemple, vous allez sélectionner les informations concernant le T-shirt de l’entité *Product* via la vue *Browse_items_Product* créée dans le **Studio Talend**. Chaque enregistrement de l’entité contient les détails définis comme critères de filtre : *Id, Name, Description* et *Price* (ID, Nom, Description et Prix).
• A partir de la Palette, déposez un composant `tMDMViewSearch` et un `tLogRow` dans l’espace de modélisation graphique.

• Reliez les composants à l’aide d’un lien `Row > Main`.

• Double-cliquez sur le `tMDMViewSearch` afin d’ouvrir sa vue `Basic settings`, dans l’onglet `Component` et de définir les propriétés du composant.

• Dans la liste `Schema`, sélectionnez `Built-In` puis cliquez sur le bouton [...] à côté du champ `Edit schema` pour ouvrir une boîte de dialogue dans laquelle vous pouvez définir la structure des données XML que vous souhaitez écrire.

• Cliquez sur le bouton `[+]` pour ajouter une colonne de type `String`. Nommez-la `Tshirt`.

• Cliquez sur `OK` pour valider la modification et passer à l’étape suivante.
Dans le champ **XML Field**, sélectionnez *Tshirt* comme colonne dans laquelle vous souhaitez écrire les données récupérées.

Dans le champ **URL**, saisissez l’adresse de votre serveur MDM et renseignez respectivement votre identifiant et votre mot de passe de connexion dans les champs **Username** et **Password**. Dans cet exemple, utilisez l’URL par défaut, puis saisissez *admin* comme identifiant et mot de passe.

Dans le champ **Data Container**, saisissez le nom du conteneur de données : *Product*.

Dans le champ **View Name**, saisissez le nom de la vue : *Browse_item_Product*.

Sous le tableau **Operations**, cliquez sur le bouton [+ ] pour ajouter une ligne.

Dans le tableau **Operations**, définissez la colonne **XPath** comme *Product/Name*, ce qui signifie que l’opération de filtre sera effectuée sur le noeud *Name*. Sélectionnez ensuite **Contains** dans la colonne **Function** puis saisissez *Tshirt* dans la colonne **Value**.

Sous le tableau **Order (One Row)**, cliquez sur le bouton [+ ] afin d’ajouter une ligne.

Dans le tableau **Order (One Row)**, définissez la colonne **XPath** comme *Product/Id* et sélectionnez l’ordre *asc* dans la colonne **Order**.

Dans l’espace de modélisation graphique, cliquez sur le tLogRow afin d’ouvrir sa vue **Basic settings** et définir ses propriétés.

![Image de la fenêtre de tLogRow](image.png)

A côté du bouton [...] utilisé pour éditer le schéma, cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.

Appuyez sur **F6** afin d’exécuter le Job.
Dans la console de la vue Run, vous pouvez voir les enregistrements Tshirt retournés, avec leur structure XML, triés en ordre ascendant.
tMemorizeRows

Ce composant mémorise une séquence de lignes passant dans ce composant et permet au(x) composant(s) suivant(s) d’effectuer des opérations de votre choix sur les lignes mémorisées.

Le tMemorizeRows mémorise temporairement une collection de données entrantes en séquence ligne par ligne et instancie cette collection en indexant chaque ligne mémorisée à partir de 0. Le nombre maximum de lignes à mémoriser à un moment donné est défini dans la vue Basic settings.

Propriétés du tMemorizeRows Standard

Ces propriétés sont utilisées pour configurer le tMemorizeRows s’exécutant dans le framework de Jobs Standard.

Le composant tMemorizeRows Standard appartient à la famille Misc.

Le composant de ce framework est toujours disponible.

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en tant qu’étape intermédiaire dans un flux de données ou comme dernière étape avant de lancer un sous-job. Note: Vous pouvez utiliser la variable globale NB_LINE_ROWS afin de récupérer la valeur renseignée |
Scénario : Récupérer les différents âges et l'âge le plus bas

Ce scénario affiche le nombre d'occurrences des différents âges et l'âge le plus bas d'un groupe de clients. Dans ce scénario, les données client sont saisies manuellement.

Vous allez voir découvrir deux manières de gérer ces données via le tMemorizeRows :

- au sein du même sous-job (avec un tJavaFlex)
- hors du sous-job du tMemorizeRows (avec un tJava)

Ce Job utilise les cinq composants suivants :

- tFixedFlowInput : ce composant contient les lignes de données client, notamment les IDs, noms et âges des clients.
- tSortRow : ce composant trie les lignes selon les données de l'âge.
- tMemorizeRows : il mémorise temporairement un nombre spécifique de lignes de données entrantes à un moment donné et indexe les lignes de données mémorisées.
- tJavaFlex : ce composant compare la valeur des âges dans les données mémorisées par le composant précédent, compte le nombre d'occurrences des différents âges et affiche ces âges dans la vue Run.
- tJava : il affiche le nombre d'occurrences des différents âges, ainsi que l'âge le plus bas.
Pour reproduire le scénario, procédez comme dans les sections suivantes.

**Déposer et relier les composants**

**Procédure**

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : `tFixedFlowInput`, `tSortRow`, `tMemorizeRows`, `tJavaFlex` et `tJava` en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le `tFixedFlowInput` au `tSortRow`, à l'aide d'un lien `Row > Main`.
3. Répétez l'opération pour connecter le `tSortRow` au `tMemorizeRows` et le `tMemorizeRows` au `tJavaFlex` à l'aide d’un lien `Row > Main`.

**Configurer les composants**

**Configurer le composant tFixedFlowInput**

**Procédure**

1. Double-cliquez sur le composant `tFixedFlowInput` afin d'ouvrir sa vue `Basic settings`.

   ![Diagramme de configuration de tFixedFlowInput](image)
   
3. Dans cet éditeur, cliquez trois fois sur le bouton [+] pour ajouter trois colonnes, que vous nommerez : id, name et age.

4. Dans la colonne Type, sélectionnez Integer pour les colonnes id et age.

5. Cliquez sur OK pour fermer l’éditeur, cliquez sur Yes afin de valider ces modifications et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

6. Sélectionnez Use Inline Content (delimited file) dans la zone Mode.

Dans le champ Content, saisissez les données clients suivantes :

<table>
<thead>
<tr>
<th></th>
<th>1; Judy; 27</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2; Lily; 45</td>
</tr>
<tr>
<td></td>
<td>3; Peter; 59</td>
</tr>
<tr>
<td></td>
<td>4; John; 30</td>
</tr>
<tr>
<td></td>
<td>5; Teddy; 45</td>
</tr>
</tbody>
</table>

Configurer le composant tSortRow

Procédure

1. Double-cliquez sur le tSortRow afin d’ouvrir sa vue Component.

2. Dans le tableau Criteria, cliquez sur le bouton [+] afin d’ajouter une ligne.

3. Dans la colonne Schema column, sélectionnez la colonne de données sur laquelle vous souhaitez baser le tri. Dans cet exemple, sélectionnez age, puisque les âges seront comparés et comptés.

4. Dans la colonne sort num or alpha?, sélectionnez le type de tri à effectuer. Dans cet exemple, comme age contient des données de type integer, sélectionnez num, pour numérique.

5. Dans la colonne Order asc or desc?, sélectionnez desc pour afficher les données en ordre descendant dans la console.

Configurer le composant tMemorizeRows

Procédure

1. Double-cliquez sur le composant tMemorizeRows afin d’ouvrir sa vue Component.
2. Dans le champ **Row count to memorize**, saisissez le nombre maximum de lignes à mémoriser à un moment donné. Puisque vous devez comparer les âges de deux clients à chaque fois, saisissez 2. Ce composant mémorise deux lignes au maximum à un moment donné et indexe toujours la nouvelle ligne entrante comme 0 et la ligne précédente comme 1.

3. Dans la colonne **Memorize** du tableau **Columns to memorize**, cochez la (les) case(s) de la (des) colonne(s) à mémoriser. Dans cet exemple, cochez la case correspondant à la colonne *age*.

### Configurer les composants tJavaFlex et tJava

#### Procédure

1. Double-cliquez sur le **tJavaFlex** afin d’ouvrir sa vue **Component**.

2. Dans la zone **Start code**, saisissez le code Java qui sera appelé durant la phase d’initialisation. Dans cet exemple, saisissez `int count=0;` afin de déclarer une variable `count` et de lui assigner la valeur 0.

3. Dans la zone **Main code**, saisissez le code Java à appliquer à chaque ligne du flux de données. Dans ce scénario, saisissez :

```java
if(!age_tMemorizeRows_1[0].equals(age_tMemorizeRows_1[1]))
{
    count++;  
}
System.out.println(age_tMemorizeRows_1[0]);
```
Ce code compare les deux âges mémorisés à chaque fois par le tMemorizeRows et compte un changement à chaque fois que des âges différents sont trouvés. Ce code affiche également les âges indexés comme 0 par le tMemorizeRows. Lorsque le composant tJavaFlex est dans le même flux que le tMemorizeRows, le format de la variable est ColumnName_ComponentName[index].

4. Dans la zone End code, saisissez le code qui sera appelé durant la phase de fermeture. Dans cet exemple, saisissez globalMap.put("number", count); afin d'initialiser la variable globale number avec la valeur de la variable count.

5. Double-cliquez sur le tJava afin d'ouvrir sa vue Component.

6. Dans le champ Code, saisissez le code suivant pour afficher le nombre d'occurrences des différents âges et de l'âge le plus bas, parmi les données clients dans la console :System.out.println("Different ages: "+globalMap.get("number")); System.out.println("Lowest age: "+((Integer[])globalMap.get("tMemorizeRows_1_age"))[0]);

La méthode globalMap.get() est utilisée par le tJava afin de récupérer les valeurs du tableau. Notez, ici, que le tJava est utilisé en-dehors du sous-job du tMemorizeRows, le format de la variable doit être ComponentName_ColumnName, différent du format de la variable utilisée par un composant placé dans le même flux.

Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur les touches Ctrl+S pour sauvegarder votre Job.
2. Appuyez sur F6 afin d'exécuter le Job, ou cliquez sur le bouton Run de la vue Run.

Résultats

Dans la console, vous pouvez constater qu'il y a dix âges différents dans le groupe des clients et que l'âge inférieur est 27.
tMemSQLClose

Ce composant ferme une connexion à une base de données MemSQL.

Propriétés du tMemSQLClose Standard

Ces propriétés sont utilisées pour configurer le tMemSQLClose s’exécutant dans le framework de Jobs Standard.

Le composant tMemSQLClose Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tMemSQLConnection dans la liste si plus d’une connexion est attendue dans le Job.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
|                  | Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
|                  | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant s’utilise avec d’autres composants MemSQL, notamment avec le tMemSQLConnection.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour plus d’informations concernant l’usage de ce composant, consultez Scénario : Ecrire des données dans et lire des données d’une table d’une base de données MemSQL à la page 2360.
tMemSQLConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Propriétés du tMemSQLConnection Standard

Ces propriétés sont utilisées pour configurer le tMemSQLConnection s'exécutant dans le framework de Jobs Standard.

Le composant tMemSQLConnection Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Remarque :**

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut-être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier, dans le référentiel, où sont stockées les propriétés. Les champs suivants sont automatiquement renseignés à l'aide des données récupérées.</td>
</tr>
<tr>
<td>Host</td>
<td>Saisissez l'emplacement du serveur de la base de données à utiliser.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d'écoute du serveur de la base de données à utiliser.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données à utiliser.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez respectivement l'identifiant et le mot de passe de l'utilisateur.</td>
</tr>
<tr>
<td>Use or register a shared DB Connection</td>
<td>Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ <strong>Shared DB Connection Name</strong> qui s'affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de</td>
</tr>
</tbody>
</table>
données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options Use dynamic job et Use an independent process to run subjob du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.

**Advanced settings**

**Additional JDBC parameters**
Spécifiez des paramètres de connexion supplémentaires pour la connexion à la base de données en cours de création.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

**Global Variables**

**Global Variables**

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

**Règle d’utilisation**
Ce composant est généralement utilisé avec d’autres composants MemSQL.

**Limitation**
Scénario associé

Pour plus d’informations concernant l’usage de ce composant, consultez Scénario : Ecrire des données dans et lire des données d’une table d’une base de données MemSQL à la page 2360.
tMemSQLInput

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma.

Le composant tMemSQLInput se connecte à une base de données et en extrait les champs selon une requête utilisateur. Les champs récupérés sont ensuite transmis au composant suivant via une connexion de flux (Main row).

Propriétés du tMemSQLInput Standard

Ces propriétés sont utilisées pour configurer le tMemSQLInput s’exécutant dans le framework de Jobs Standard.

Le composant tMemSQLInput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut-être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier, dans le référentiel, où sont stockées les propriétés. Les champs suivants sont automatiquement renseignés à l’aide des données récupérées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>Host</th>
<th>Saisissez l’emplacement du serveur de la base de données MemSQL à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute du serveur de la base de données MemSQL à utiliser.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données MemSQL à utiliser.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez respectivement l’identifiant et le mot de passe de connexion à la base de données MemSQL à utiliser.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <code>line</code> lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <strong>No</strong> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre <strong>[Repository Content]</strong>.</td>
</tr>
<tr>
<td>Table Name</td>
<td>Nom de la table à lire.</td>
</tr>
</tbody>
</table>
### Query type et Query
Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

### Advanced settings

#### Additional JDBC parameters
Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans l’onglet **Basic settings**.

**Remarque :**

Lorsque vous devez traiter des données au format date/heure 0000-00-00 00:00:00 utilisant ce composant, définissez les paramètres comme suit :

```
noDatet imeStringSync=true&zero-Da
timeBehavior=convertToNull.
```

#### Trim all the String/Char columns
Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.

#### Trim column
Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.

**Remarque :**

Décochez **Trim all the String/Char columns** pour activer le tableau **Trim column**.

#### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

#### Global Variables
**NB_LINE** : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

**QUERY** : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données MemSQL.</th>
</tr>
</thead>
</table>

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

**Limitation**

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

**Scénario : Ecrire des données dans et lire des données d’une table d’une base de données MemSQL**

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario présente un Job construit pour écrire une table dans une base de données MemSQL et récupérer les données de la table.
Construire le Job

Pourquoi et quand exécuter cette tâche

Procedura

1. Déposez un tMemSQLConnection, un tRowGenerator, un tMemSQLInput, un tMemSQLOutput, un tMemSQLClose et un tLogRow de la Palette dans l’espace de modélisation graphique.
2. Reliez le tMemSQLConnection au tRowGenerator à l’aide d’un lien Row > OnSubjobOk.
3. Reliez le tRowGenerator au tMemSQLOutput à l’aide d’un lien Row > Main.
4. Connectez le tRowGenerator au tMemSQLInput à l’aide d’un lien Row > OnSubjobOk.
5. Reliez le tMemSQLInput au tLogRow à l’aide d’un lien Row > Main.
6. Reliez le tMemSQLInput au tMemSQLClose à l’aide d’un lien Row > OnSubjobOk.

Configurer les composants

Ouvrir et fermer la connexion à la base de données

Procedura

1. Double-cliquez sur le composant tMemSQLConnection pour ouvrir sa vue Basic settings.
2. Dans la liste **Property Type**, sélectionnez **Built-In**. Spécifiez les informations de connexion dans les champs correspondants, notamment le nom de l’hôte et le numéro du port d’écoute du serveur de la base de données, l’identifiant et le mot de passe de connexion à la base de données, ainsi que le nom de la base de données.

3. Double-cliquez sur le **tMemSQLClose** pour ouvrir sa vue **Basic settings**.

4. Dans la liste **Component List**, le composant **tMemSQLConnection_1** est automatiquement sélectionné.

**Ecrire des données dans une table de base de données**

**Procédure**

1. Double-cliquez sur le **tRowGenerator** pour ouvrir son éditeur de génération de lignes.
2. Ajoutez trois colonnes comme ci-dessous, pour générer 10 lignes de données :


- **firstName**, de type *String*. Dans sa liste *Function*, sélectionnez *TalendDataGenerator.getFirst Name* pour générer des prénoms aléatoires.

- **lastName**, de type *String*. Dans sa liste *Function*, sélectionnez *TalendDataGenerator.getLast Name* pour générer des noms de famille aléatoires.

3. Dans la vue *Basic settings* du *tRowGenerator*, cliquez sur le bouton *Edit Schema* pour ouvrir l’éditeur du schéma. Dans la colonne *Length*, saisissez respectivement 4, 10 et 10 pour les colonnes *id, firstName* et *lastName*. 
Cela fait, cliquez sur OK pour fermer l’éditeur et cliquez sur Yes lorsqu’une fenêtre vous propose de propager le schéma au composant suivant.
Pour plus d’informations concernant la configuration du tRowGenerator, consultez tRowGenerator à la page 3478.

4. Double-cliquez sur le composant tMemSQLOutput pour ouvrir sa vue Basic settings.

5. Cochez la case Use Existing Connection. Dans la liste Component List, le composant tMemSQLConnection_1 est automatiquement sélectionné.
Renseignez le champ Table en saisissant le nom de la table de base de données, customers dans cet exemple.
Sélectionnez Drop table if exists and create dans la liste Action on table et Insert dans la liste Action on table.

Lire des données d’une table d’une base de données

Procédure
1. Double-cliquez sur le tMemSQLInput pour ouvrir sa vue Basic settings.
2. Cochez la case **Use Existing Connection**. Dans la liste **Component List**, le composant `tMemSQLConnection_1` est automatiquement sélectionné.

3. Dans la liste **Schema**, sélectionnez **Built-in** et cliquez sur **Edit schema** pour définir la structure des données de la table de base de données de laquelle lire les données.

   Cliquez trois fois sur le bouton `[+]` pour ajouter les lignes à utiliser pour définir le schéma, respectivement `id`, `firstName` et `lastName`.

   Sous la colonne **Column**, cliquez dans chaque champ pour saisir le nom des colonnes.

   Cliquez dans la colonne **DB Type** pour définir le type de données.

   Cliquez sur **OK** pour fermer l’éditeur du schéma.

4. Renseignez le champ **Table** en saisissant le nom de la table de base de données, `customers` dans cet exemple.

   Cliquez sur le bouton `[...]` à côté du champ **Table Name** pour sélectionner la table de base de données souhaitée. Une boîte de dialogue s’ouvre et affiche une arborescence de toutes les tables de la base de données sélectionnée. Sélectionnez `customers` et cliquez sur **OK** pour fermer la boîte de dialogue.
5. Dans la liste **Query Type**, sélectionnez **Built-In**. Cliquez sur le bouton **Guess Query**. Le champ **Query** est automatiquement renseigné et récupère toutes les colonnes de la table sélectionnée.

6. Double-cliquez sur le **tLogRow** pour afficher sa vue **Basic settings**.

7. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

8. Sauvegardez le Job.

**Exécuter le Job**

Appuyez sur **F6** pour afficher les résultats dans la console.
Les données, avec les noms de colonnes souhaités, sont récupérées de la table de base de données.
tMemSQLOutput

Ce composant lit des données provenant du composant précédent dans le Job et exécute l’action définie sur une table d’une base de données MemSQL et/ou sur les données contenues dans la table.
Le tMemSQLOutput se connecte à une base de données MemSQL et écrit, met à jour, modifie ou supprime des entrées dans la base de données.

Propriétés du tMemSQLOutput Standard

Ces propriétés sont utilisées pour configurer le tMemSQLOutput s’exécutant dans le framework de Jobs Standard.
Le composant tMemSQLOutput Standard appartient aux familles Big Data et Databases.
Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Remarque :</td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td>1.</td>
<td>Au niveau parent, enregistrer la connexion à la base de données à partager, dans</td>
</tr>
</tbody>
</table>
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Saisissez l’emplacement du serveur de la base de données MemSQL à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute du serveur de la base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données MemSQL à utiliser.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les données d’authentification afin de vous connecter à la base de données à utiliser.</td>
</tr>
<tr>
<td>Table</td>
<td>Saisissez le nom de la table à écrire. Notez que seule une table peut être écrite à la fois.</td>
</tr>
</tbody>
</table>

**Action on table**

Sur la table définie, vous pouvez effectuer l’une des opérations suivantes :

- **None** : aucune opération n’est effectuée.
- **Drop and create a table** : la table est supprimée et créée à nouveau.
- **Create a table** : la table n’existe pas et est créée.
- **Create a table if not exists** : la table est créée si elle n’existe pas.
- **Drop a table if exists and create** : la table est supprimée si elle existe déjà et est créée à nouveau.
- **Clear a table** : le contenu de la table est supprimé.

**Action on data**

Sur les données de la table définie, vous pouvez effectuer les actions suivantes :

- **Insert** : ajouter de nouvelles entrées à la table. S’il y a des doublons, le Job s’arrête.
- **Update** : modifier les entrées existantes.
- **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
- **Delete** : supprimer les entrées correspondant au flux d’entrée.

⚠️ **Avertissement** :

<table>
<thead>
<tr>
<th><strong>Schema et Edit schema</strong></th>
<th>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-In</strong></td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de proposer ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
</tbody>
</table>
### Advanced settings

<table>
<thead>
<tr>
<th><strong>Die on error</strong></th>
<th>Cette case est décochée par défaut. Cela vous permet d’ignorer les lignes en erreur et de terminer le traitement des lignes sans erreur.</th>
</tr>
</thead>
</table>

#### Additional JDBC parameters

Spécifiez les propriétés de connexion supplémentaires pour la connexion à la base de données en cours de création. Cette option n’est pas disponible si vous avez coché la case *Use an existing connection* dans l’onglet *Basic settings*.

**Remarque :**

Vous pouvez appuyer sur les touches **Ctrl + Espace** pour accéder à une liste de variables globales prédéfinies.

<table>
<thead>
<tr>
<th><strong>Use batch size</strong></th>
<th>Cochez cette case pour activer le mode de traitement de données par lot. Dans le champ <em>Batch Size</em> qui apparaît lorsque cette case est cochée, saisissez le nombre représentant la taille du lot à traiter.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Additional Columns</strong></th>
<th>Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Saisissez le nom de la colonne du schéma à modifier ou insérer.</td>
</tr>
<tr>
<td><strong>SQL expression</strong></td>
<td>Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.</td>
</tr>
<tr>
<td><strong>Position</strong></td>
<td>Sélectionnez <em>Before</em>, <em>Replace</em> ou <em>After</em>, en fonction de l’action à effectuer sur la colonne de référence.</td>
</tr>
<tr>
<td><strong>Reference column</strong></td>
<td>Saisissez une colonne de référence que le composant <em>tJDBCOutput</em> peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Use field options</strong></th>
<th>Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Enable debug mode</strong></th>
<th>Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>tStatCatcher Statistics</strong></th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>
**Global Variables**

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier. |
| | **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable *After* et retourne un entier. |
| | **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable *After* et retourne un entier. |
| | **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable *After* et retourne un entier. |
| | **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable *After* et retourne un entier. |
| | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. |

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le [Guide utilisateur du Studio Talend](#).

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

---

**Utilisation**

| Règle d’utilisation | Ce composant doit être utilisé comme composant de sortie. Il vous permet d’effectuer des actions sur une table d’une base de données MemSQL. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue **Basic settings**. Lorsqu’un paramètre
dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Limitation</th>
</tr>
</thead>
</table>

**Scénario associé**

Pour plus d’informations concernant l’usage de ce composant, consultez **Scénario : Ecrire des données dans et lire des données d’une table d’une base de données MemSQL** à la page 2360.
tMemSQLRow

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler).

Le tMemSQLRow exécute des requêtes SQL déclarées sur la base de données spécifiée, selon la nature de la requête et de la base de données. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Propriétés du tMemSQLRow Standard

Ces propriétés sont utilisées pour configurer le tMemSQLRow s'exécutant dans le framework de Jobs Standard.

Le composant tMemSQLRow Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

| Host | Saisissez l’emplacement du serveur de la base de données MemSQL à utiliser. |
| Port | Saisissez le numéro du port d’écoute du serveur de la base de données. |
| Database | Saisissez le nom de la base de données MemSQL à utiliser. |
| Username et Password | Saisissez les données d’authentification afin de vous connecter à la base de données à utiliser. |
| Table Name | Nom de la table à traiter. |

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th>Query type</th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong>: Renseignez manuellement l’instruction de requête ou construisez-la graphiquement à l’aide du SQLBuilder.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong>: Sélectionnez la requête appropriée dans le <strong>Repository</strong>. Le champ <strong>Query</strong> est renseigné automatiquement.</td>
</tr>
<tr>
<td><strong>Guess Query</strong></td>
<td>Cliquez sur le bouton <strong>Guess Query</strong> pour générer la requête correspondant au schéma de la table dans le champ <strong>Query</strong>.</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cette case est décochée par défaut. Cela vous permet d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur. Si nécessaire, vous pouvez récupérer les lignes en erreur via un lien <strong>Row &gt; Rejects</strong>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| **Additional JDBC parameters** | Spécifiez les propriétés de connexion supplémentaires pour la connexion à la base de données en cours de création. Cette option n’est pas disponible si vous avez coché la case **Use an existing connection** dans l’onglet **Basic settings**. |
| **Propagate QUERY’s record set** | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. |

**Remarque** :

Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. La colonne contenant le résultat de la requête doit être de type **Object**. Ce composant est généralement suivi du **tParseRecordSet**.
représentés par des ‘?’ dans l'instruction SQL définie dans le champ Query de l'onglet **Basic settings**.

**Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL.

**Parameter Type** : Saisissez le type du paramètre.

**Parameter Value** : Saisissez la valeur du paramètre.

**Remarque** :
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

#### Global Variables

**QUERY** : requête traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

#### Règle d'utilisation
Ce composant couvre toutes les possibilités de requête SQL dans les bases de données MemSQL.

#### Dynamic settings
Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Limitation**

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center ([https://help.talend.com](https://help.talend.com)).

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tMicrosoftCrmInput

Ce composant extrait les données d’une base Microsoft CRM selon certaines conditions définies dans des colonnes spécifiques.

**Propriétés du tMicrosoftCrmInput Standard**

Ces propriétés sont utilisées pour configurer le tMicrosoftCrmInput s’exécutant dans le framework de Jobs Standard.

Le composant tMicrosoftCrmInput Standard appartient à la famille Business.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Type</td>
<td>Sélectionnez le type d’authentification correspondant à votre modèle de déploiement spécifique de votre CRM.</td>
</tr>
<tr>
<td></td>
<td>• <strong>ON PREMISE</strong> : le CRM est installé sur site sur les serveurs du client et la licence est achetée avec toute infrastructure nécessaire.</td>
</tr>
<tr>
<td></td>
<td>• <strong>ONLINE</strong> : le client accède au CRM en tant que partie du service basé Cloud nécessitant souscription, hébergé par Microsoft dans leurs centres de données.</td>
</tr>
<tr>
<td>Registered application type</td>
<td>Sélectionnez le type de votre application, Native App ou Web App with delegated permissions.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété est disponible pour Microsoft CRM ONLINE 2016.</td>
</tr>
<tr>
<td>Microsoft WebService URL</td>
<td>Spécifiez l’URL du Service Web d’un serveur Microsoft CRM sur site.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2011.</td>
</tr>
<tr>
<td>OrganizeName</td>
<td>Spécifiez le nom de l’entreprise nécessitant l’accès à la base de données Microsoft CRM.</td>
</tr>
<tr>
<td>Propriété</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Spécifiez les informations d’authentification de l’utilisateur requises pour accéder à la base de données Microsoft CRM. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Domain</td>
<td>Spécifiez le nom de domaine du serveur sur lequel est hébergé la base de données Microsoft CRM. Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2011 et 2016.</td>
</tr>
<tr>
<td>Port</td>
<td>Spécifiez le numéro du port d’écoute du serveur de la base de données Microsoft CRM. Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2011.</td>
</tr>
<tr>
<td>Discovery service endpoint</td>
<td>Spécifiez l’endpoint du service de découverte du CRM. Cette propriété est disponible pour Microsoft CRM ONLINE 2011.</td>
</tr>
<tr>
<td>Service Root URL</td>
<td>Spécifiez le nombre de secondes à attendre un message de réponse du serveur du CRM avant d’en fermer la connexion. Cette propriété est disponible pour Microsoft CRM ONLINE 2016.</td>
</tr>
<tr>
<td>Application ID</td>
<td>Spécifiez l’identifiant client unique qu’Azure AD assigne à votre application lorsque vous enregistrez l’application dans le portail Azure. Pour plus d’informations, consultez <a href="#">Enregistrer une application Dynamics 365 auprès d’Azure Active Directory</a>.</td>
</tr>
</tbody>
</table>
## Application secret

Clé d’accès à votre application Web. Pour plus d’informations concernant l’obtention de cette valeur, consultez l’article [Obtenir la clé d’accès](#).

*This property is available when Web App with delegated permissions is selected from the Registered application type drop-down list.*

## OAuth authorization endpoint

Spécifiez l’endpoint d’autorisation OAuth à utiliser pour la découverte OAuth. Pour plus d’informations, consultez l’article [Points de terminaison d’autorisation OAuth](#).

Cette propriété est disponible pour Microsoft CRM ONLINE 2016.

## Time out (second)

Spécifiez le nombre de secondes à attendre un message de réponse du serveur du CRM avant d’en fermer la connexion.

## Entity/EntitySet

Sélectionnez l’entité/le jeu d’entités correspondant(e) dans la liste, ou sélectionnez CustomEntity/CustomEntitySet dans la liste. Puis, dans le champ Custom Entity Name qui s’affiche, spécifiez le nom de l’entité/du jeu d’entités et définissez le schéma pour l’entité personnalisée/le jeu d’entités personnalisées.

## Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In**: Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le [Guide utilisateur du Studio Talend](#).
- **Repository**: Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le [Guide utilisateur du Studio Talend](#).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content](#).
Le schéma de ce composant est relatif à l'entité spécifiée.

### Custom filter
Cochez cette case et, dans le champ qui s’affiche, spécifiez les critères de recherche utilisant les fonctions standard de la requête OData, comme contains, endswith et startswith. Pour plus d’informations concernant les fonctions de requêtes supportées, consultez la description des fonctions des requêtes Standard à la page [Query Data using the Web API](en anglais).

Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2016 et ONLINE 2016.

### Logical operator used to combine conditions
Sélectionnez l’opérateur logique utilisé pour combiner les conditions définies dans la table **Conditions**.

- **And** : l’enregistrement est inclus dans l’ensemble des résultats lorsque toutes les conditions sont respectées.
- **Or** : l’enregistrement est inclus dans l’ensemble des résultats lorsqu’une des conditions est respectée.

### Conditions
Cliquez sur le bouton **[+]** pour ajouter autant de conditions que nécessaire et spécifiez les paramètres suivants pour chaque condition.

- **Input column** : Cliquez dans la cellule et sélectionnez la colonne du schéma d’entrée à laquelle attribuer la condition.
- **Operator** : Cliquez dans la cellule et sélectionnez l’opérateur utilisé pour comparer la valeur de la colonne d’entrée à la valeur définie dans la colonne **Value**. La valeur de ce paramètre peut être Equal, NotEqual, GreaterThan, LessThan, GreaterEqual, LessEqual et Like.
- **Value** : Saisissez la valeur entre guillemets, si nécessaire.

### Order by
Cochez cette case et, dans la table qui s’affiche, spécifiez les colonnes en vous basant sur les données récupérées qui seront triées en ordre ascendant ou descendant. Plusieurs colonnes de tri peuvent être spécifiées et la hiérarchie du tri se base sur l’ordre des colonnes spécifiées. Cela signifie que les données récupérées seront triées par la première colonne et que la liste triée est triée à son tour par la deuxième colonne et ainsi de suite.

Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2016 et ONLINE 2016.

### Advanced settings

<table>
<thead>
<tr>
<th>Reuse Http Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour maintenir la connexion en cours. Décrochez-la pour libérer la connexion.</td>
</tr>
</tbody>
</table>

2382
Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2015.

### Max number of reconnection attempts

Spécifiez le nombre maximal de tentatives de reconnexion à la base de données Microsoft CRM après l’expiration du jeton. Si cette valeur est définie à 0 ou inférieure à 0, vous ne pourrez pas vous reconnecter.

Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2016, ONLINE 2011 et ONLINE 2016.

### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Régie d’utilisation</td>
<td>Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie.</td>
</tr>
</tbody>
</table>

### Scénario : Écrire des données dans une base de données Microsoft CRM et attribuer des conditions aux colonnes pour extraire des lignes spécifiques

Le scénario suivant décrit un Job à quatre composants qui a pour but l’écriture des données d’un fichier délimité en entrée dans une entité personnalisée d’une base de données Microsoft tCRM. Le Job permet ensuite d’extraire des lignes spécifiques jusque dans un fichier de sortie, en utilisant les conditions définies sur certaines colonnes d’entrée.

⚠️ **Avertissement :**

Si vous voulez écrire dans une Entité Personnalisée d’une base de données Microsoft CRM, assurez-vous de nommer les colonnes conformément aux règles de syntaxe Microsoft, à savoir "name_columnname" en minuscules.
Configurer le Job

Procédure

1. A partir de la Palette, déposez les composants suivants dans l'éditeur graphique : tFileInputDelimited, tFileOutputDelimited, tMicrosoftCrmInput, et tMicrosoftCrmOutput.

![Diagramme des composants](image)

2. Connectez le composant tFileInputDelimited au tMicrosoftCrmOutput à l'aide d'un lien de type Row > Main.
3. Connectez le composant tMicrosoftCrmInput au tFileOutputDelimited à l'aide d'un lien de type Row > Main.
4. Connectez le composant tFileInputDelimited au tMicrosoftCrmInput à l'aide d'un lien de type Trigger > OnSubjobOk.

Configurer le tFileInputDelimited

Procédure

1. Double-cliquez sur le composant tFileInputDelimited pour afficher la vue Basic settings et configurer ses paramètres de base.

![Vignette de Basic settings](image)

2. Sélectionnez le mode Repository à partir de la liste déroulante Property Type si vous avez stocké les paramètres du fichier d'entrée sous le nœud Metadata de l'arborescence Repository. Dans le cas contraire, sélectionnez le mode Built-In et remplissez manuellement les champs correspondants. Dans cet exemple, les propriétés sont définies en mode Built-In.
3. Cliquez sur le bouton [...] à côté du champ File name/Stream et parcourez vos dossiers jusqu'au fichier délimité contenant les données d'entrée. Dans cet exemple, le fichier d'entrée comprend
les colonnes suivantes : new_id, new_status, new_firstname, new_email, new_city, new_initial et new_zipcode.

4. Dans le champ **Row separator**, définissez le séparateur de lignes permettant d'identifier les fins de ligne. De même, pour **Field separator**, définissez le séparateur de champs permettant de délimiter les différents champs pour chaque ligne.

5. Définissez, si nécessaire, le nombre de lignes d'en-tête (**Header**) et de pied de page (**Footer**) à ignorer, ainsi que le nombre maximum de lignes (**Limit**) que vous souhaitez traiter. Dans cet exemple, aucun de ces trois champs ne sont définis.

6. Cliquez sur **Edit schema** pour ouvrir une boîte de dialogue dans laquelle vous pouvez définir le schéma d’entrée que vous voulez écrire dans la base de données MicrosoftCRM.

7. Cliquez sur **OK** pour fermer la boîte de dialogue.

**Configurer le tMicrosoftCrmOutput**

**Procédure**

1. Double-cliquez sur le composant **tMicrosoftCrmOutput** pour afficher la vue **Basic settings** et configurer ses paramètres de base.
2. Dans le champ **Microsoft Web Service URL**, saisissez l’URL du serveur Web Microsoft. Renseignez ensuite le nom (**Username**) ainsi que le mot de passe (**Password**) correspondants.

3. Dans le champ **OrganizeName**, saisissez l’identifiant qui porte les droits d’accès à la base de données MicrosoftCRM.

4. Dans le champ **Domain**, renseignez le nom de domaine du serveur sur lequel la base de données MicrosoftCRM est hébergée. Saisissez ensuite l’adresse IP (**Host**) et le numéro de port d’écoute (**Port**) du serveur.

5. A partir de la liste déroulante **Action**, sélectionnez l’opération que vous voulez effectuer. Dans cet exemple, le but est d’insérer (**Insert**) des données dans une entité personnalisée sur la base MicrosoftCRM.

6. Dans le champ **Time out**, définissez la limite de temps (en secondes) après laquelle le Job s’arrête automatiquement.

7. Dans la liste **Entity**, sélectionnez l’une des options proposées. Dans cet exemple, c’est l’option **CustomEntity** qui est sélectionnée.

   **Remarque :**

   Lorsque l’option **CustomEntity** est sélectionnée, le champ **Custom Entity Name** apparaît, dans lequel vous devez saisir le nom de l’entité personnalisée.

   Le **Schema** est alors défini de manière automatique, en fonction de l’entité sélectionnée. Cliquez si nécessaire sur le bouton **Edit schema** pour afficher une boîte de dialogue à partir de laquelle vous pouvez modifier ce schéma et retirer les colonnes dont vous n’avez pas besoin en sortie.

8. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
Configurer le tMicrosoftCrmInput

Procédure

1. Double-cliquez sur le composant tMicrosoftCrmInput pour afficher la vue Basic settings et configurer ses paramètres de base.

2. Sélectionnez le mode Repository à partir de la liste déroulante Property Type si vous avez stocké les paramètres du fichier d’entrée sous le nœud Metadata de l’arborescence Repository. Dans le cas contraire, sélectionnez le mode Built-In et remplissez manuellement les champs correspondants. Dans cet exemple, les propriétés sont définies en mode Built-In.

3. Dans le champ Microsoft Web Service URL, saisissez l’URL du serveur Web Microsoft. Renseignez ensuite le nom (Username) ainsi que le mot de passe (Password) correspondants.

4. Dans le champ OrganizeName, saisissez l’identifiant qui porte les droits d’accès à la base de données MicrosoftCRM.

5. Dans le champ Domain, renseignez le nom de domaine du serveur sur lequel la base de données MicrosoftCRM est hébergée. Saisissez ensuite l’adresse IP (Host) et le numéro du port d’écoute (Port) du serveur.

6. Dans la liste Entity, sélectionnez l’une des options proposées. Dans cet exemple, c’est l’option CustomEntity qui est sélectionnée.

7. Le Schema est alors défini de manière automatique, en fonction de l’entité sélectionnée, mais vous pouvez le modifier si nécessaire. Dans cet exemple, pour accéder à l’entité personnalisée, il faut définir le schéma manuellement. Copiez le schéma à sept colonnes à partir du composant tMicrosoftCrmOutput et collez-le dans la boîte de dialogue du tMicrosoftCrmInput.
8. Cliquez sur le bouton OK pour fermer la boîte de dialogue. Un message pop-up vous proposera de propager les modifications : cliquez sur le bouton Yes.

9. Dans la vue Basic settings, sélectionnez l'opérateur logique And ou Or à utiliser pour combiner les conditions définies dans les colonnes d’entrée. Dans cet exemple, pour définir deux conditions sur deux colonnes d’entrée différentes, il faut utiliser l’opérateur logique And.

10. Dans la zone Condition, cliquez sur le bouton [...] pour ajouter autant de lignes que nécessaire, puis cliquez dans chaque ligne et sélectionnez, à partir de la liste déroulante Input column, la colonne sur laquelle appliquer la condition. Dans cet exemple, les deux colonnes new-city et new_id portent des conditions. Le but est d’extraire toutes les lignes concernant les clients dont la ville correspond à “New York” et dont l’identifiant est supérieur à 2.


**Configurer le tFileOutputDelimited**

**Procédure**

1. Double-cliquez sur le composant tFileOutputDelimited pour afficher la vue Basic settings et configurer ses paramètres de base.
2. Sélectionnez le mode Built-In à partir de la liste déroulante Property Type puis cliquez sur le bouton [...] à côté du champ File Name pour parcourir vos dossiers jusqu’au fichier de sortie.

3. Définissez les séparateurs de lignes (Row separator) et de champs (Field separator).

4. Cochez la case Append pour ajouter des lignes à la fin des entrées de la base de données.

5. Cochez la case Include Header si le fichier de sortie comprend un en-tête.

6. Cliquez sur le bouton Sync columns pour récupérer le schéma à partir du composant précédent.

**Exécuter votre Job**

**Procédure**

Enregistrez votre Job puis cliquez sur F6 pour l’exécuter.

**Résultats**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7;married;Harry;<a href="mailto:jnewman@comp.com">jnewman@comp.com</a>;New York;H.M;55677</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8;married;Jerry;<a href="mailto:jnewman@comp.com">jnewman@comp.com</a>;New York;J.M;55677</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9;married;Alice;<a href="mailto:anewman@comp.com">anewman@comp.com</a>;New York;A.M;55677</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10;single;Jack;<a href="mailto:jnewman@comp.com">jnewman@comp.com</a>;New York;J.M;55677</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Seuls les clients habitant la ville de New York, dont l’identifiant est supérieur à 2, apparaissent dans la liste du fichier de sortie stocké en local.
tMicrosoftCrmOutput

Ce composant écrit des données dans une base de données Microsoft CRM.

Propriétés du tMicrosoftCrmOutput Standard

Ces propriétés sont utilisées pour configurer le tMicrosoftCrmOutput s’exécutant dans le framework de Jobs Standard.

Le composant tMicrosoftCrmOutput Standard appartient à la famille Business.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Authentication Type</th>
<th>Sélectionnez le type d’authentification correspondant à votre modèle de déploiement spécifique de votre CRM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ON_PREMISE</td>
<td>le CRM est installé sur site sur les serveurs du client et la licence est achetée avec toute infrastructure nécessaire.</td>
</tr>
<tr>
<td>• ONLINE</td>
<td>le client accède au CRM en tant que partie du service basé Cloud nécessitant souscription, hébergé par Microsoft dans leurs centres de données.</td>
</tr>
</tbody>
</table>

|-------------|--------------------------------------------------------------------------------------------------|

|-------------|--------------------------------------------------------------------------------------------------|

| Registered application type | Sélectionnez le type de votre application, Native App ou Web App with delegated permissions. |
|                            | Cette propriété est disponible pour Microsoft CRM ONLINE 2016. |

<table>
<thead>
<tr>
<th>Microsoft WebService URL</th>
<th>Spécifiez l’URL du Service Web d’un serveur Microsoft CRM sur site.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2011.</td>
</tr>
</tbody>
</table>

<p>| OrganizeName | Spécifiez le nom de l’entreprise nécessitant l’accès à la base de données Microsoft CRM. |</p>
<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username et Password</td>
<td>Spécifiez les informations d’authentification de l’utilisateur requises pour accéder à la base de données Microsoft CRM. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Domain</td>
<td>Spécifiez le nom de domaine du serveur sur lequel est hébergé la base de données Microsoft CRM. Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2011 et 2016.</td>
</tr>
<tr>
<td>Port</td>
<td>Spécifiez le numéro du port d’écoute du serveur de la base de données Microsoft CRM. Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2011.</td>
</tr>
<tr>
<td>Discovery service endpoint</td>
<td>Spécifiez l’endpoint du service de découverte du CRM. Cette propriété est disponible pour Microsoft CRM ONLINE 2011.</td>
</tr>
<tr>
<td>Service Root URL</td>
<td>Spécifiez le nombre de secondes à attendre un message de réponse du serveur du CRM avant d’en fermer la connexion. Cette propriété est disponible pour Microsoft CRM ONLINE 2016 et ONLINE 2016.</td>
</tr>
<tr>
<td>Application ID</td>
<td>Spécifiez l’identifiant client unique qu’Azure AD assigne à votre application lorsque vous enregistrez l’application dans le portail Azure. Pour plus d’informations, consultez Enregistrer une application Dynamics 365 auprès d’Azure Active Directory.</td>
</tr>
</tbody>
</table>
| **Application secret** | Cette propriété est disponible pour Microsoft CRM ONLINE 2016.  
Clé d'accès à votre application Web. Pour plus d'informations concernant l'obtention de valeur, consultez.  
This property is available when Web App with delegated permissions is selected from the Registered application type drop-down list. |
|------------------------|----------------------------------------------------------------------------------------------------------|
| **OAuth authorization endpoint** | Spécifiez l'endpoint d'autorisation OAuth à utiliser pour la découverte OAuth. Pour plus d'informations, consultez Points de terminaison d'autorisation OAuth.  
Cette propriété est disponible pour Microsoft CRM ONLINE 2016. |
| **Action** | Sélectionnez dans la liste l'action que vous souhaitez effectuer sur les données du CRM, insert, update ou delete.  
| **Time out (second)** | Spécifiez le nombre de secondes à attendre un message de réponse du serveur du CRM avant d'en fermer la connexion. |
| **Entity/EntitySet** | Sélectionnez l'entité/le jeu d'entités correspondant(e) dans la liste, ou sélectionnez CustomEntity/CustomEntitySet dans la liste.  
Puis, dans le champ Custom Entity Name qui s'affiche, spécifiez le nom de l'entité/du jeu d'entités et définissez le schéma pour l'entité personnalisée/le jeu d'entités personnalisées. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma. |
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

Le schéma de ce composant est relatif à l’entité spécifiée.

### Lookup Type Mapping

Renseignez cette table pour mapper la propriété de lookup à son type d’entité associé ou à sa valeur du jeu d’entités associé.

- **Input column** : nom de la colonne du schéma de la propriété de lookup.
- **Type** : type de l’entité associée pour la propriété de lookup correspondante. Non disponible pour ON-PREMISE 2016 et ONLINE 2016.

Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2015.

### Advanced settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuse Http Client</td>
<td>Cochez cette case pour maintenir la connexion en cours. Décocochez-la pour libérer la connexion.</td>
</tr>
<tr>
<td>Transform empty lookup string values to NULL</td>
<td>Cochez cette case pour écrire la chaîne de caractères de référence vide dans la base de données Microsoft CRM en tant que valeur NULL.</td>
</tr>
<tr>
<td>Ignore Null</td>
<td>Cochez cette case pour ignorer les valeurs NULL dans les données d’entrée.</td>
</tr>
<tr>
<td>Max number of reconnection attempts</td>
<td>Spécifiez le nombre maximal de tentatives de reconnexion à la base de données Microsoft CRM après</td>
</tr>
</tbody>
</table>
l'expiration du jeton. Si cette valeur est définie à 0 ou inférieure à 0, vous ne pourrez pas vous reconnecter.
Cette propriété est disponible pour Microsoft CRM ON-PREMISE 2016, ONLINE 2011 et ONLINE 2016.

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. |

### Variables globales

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

| Règle d'utilisation | Ce composant est généralement utilisé comme composant de fin dans un Job ou un sous-job et nécessite un lien d'entrée. |

### Scénario associé

Pour un scénario associé, consultez Scénario : Écrire des données dans une base de données Microsoft CRM et attribuer des conditions aux colonnes pour extraire des lignes spécifiques à la page 2383.
tMicrosoftMQInput

Ce composant récupère le premier message d'une file d'attente de message Microsoft donnée (le composant supporte uniquement le type String).

Le tMicrosoftMQInput récupère les messages un par un dans l'ordre de leur ID, dans la file d'attente de message Microsoft. Chaque exécution récupère un seul message.

Propriétés du tMicrosoftMQInput Standard

Ces propriétés sont utilisées pour configurer le tMicrosoftMQInput s'exécutant dans le framework de Jobs Standard.

Le composant tMicrosoftMQInput Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>Toujours Built-in. Le schéma contient une colonne en lecture seule de type String nommée message.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Saisissez le nom de l'hôte ou l'adresse IP du serveur hôte.</td>
</tr>
<tr>
<td>Queue</td>
<td>Saisissez le nom de la file d'attente à partir de laquelle vous souhaitez récupérer les messages.</td>
</tr>
<tr>
<td>Queue type</td>
<td>Sélectionnez un type de file d'attente de message Microsoft dans la liste déroulante.</td>
</tr>
<tr>
<td></td>
<td>• Private : une file d'attente privée, qui est enregistrée sur la machine locale.</td>
</tr>
<tr>
<td></td>
<td>• Public : une file d'attente publique, qui est enregistrée dans le service de répertoire.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est décochée par défaut, pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant. |

Global Variables

| Global Variables | ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est |
Une variable **After** et retourne une chaine de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job. Il doit être lié à un composant de sortie.</th>
</tr>
</thead>
</table>
| **Connexions**      | Liens de sortie (de ce composant à un autre) :
|                     | **Row** : Main, Iterate.  
|                     | Liens d’entrée (d’un autre composant à celui-ci) :
|                     | **Row** : Main, Iterate  
|                     | Pour plus d’informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*. |
| **Limitation**      | Ce composant ne supporte que le type String et fonctionne uniquement sous Windows.  
|                     | Ce composant requiert l’installation des fichiers .jar liés. |

### Scénario : Ecrire et récupérer des messages dans une file d’attente de message Microsoft

Ce scénario se compose de deux Jobs. Le premier Job poste des messages dans une file de message Microsoft et le second Job récupère les messages du serveur.

**Poster des messages dans une file de messages Microsoft**

Dans le premier Job, un message de type string est créé via un tRowGenerator et posté dans une file d’attente de message Microsoft à l’aide d’un composant tMicrosoftMQOutput. Un composant tLogRow intermédiaire affiche le flux passé d’un composant à l’autre.
Vous pouvez voir que la file a été créée automatiquement et que les messages ont été écrits.

**Déposer et relier les composants**

**Procédure**

1. Déposez de la Palette dans l'espace de modélisation graphique un tRowGenerator, un tLogRow et un tMicrosoftMQOutput.

2. Connectez les composants à l'aide de liens Row > Main.

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le tRowGenerator afin d'ouvrir son éditeur.
2. Dans cet éditeur, cliquez sur le bouton [+] pour ajouter trois lignes au schéma.

3. Dans la colonne Column, saisissez un nouveau nom pour chaque ligne, afin de les renommer. Ici, saisissez respectivement ID, Name et Address.

4. Dans la colonne Type, sélectionnez Integer pour la ligne ID dans la liste déroulante et laissez les autres lignes en String.

5. Dans la colonne Functions, sélectionnez random pour la ligne ID, getFirstName pour la ligne Name et getUsCity pour la ligne Address.

6. Dans le champ Number of Rows for RowGenerator, saisissez 12 afin de limiter le nombre de lignes à générer. Puis cliquez sur OK pour valider vos modifications.

   **Remarque :**
   Lors d’une utilisation réelle, vous devrez utiliser un composant d’entrée pour charger les données qui vous intéressent, au lieu du tRowGenerator.

7. Double-cliquez sur le composant tMicrosoftMQOutput afin d’ouvrir sa vue Component.


10. Dans le champ Message column (String Type), sélectionnez Address dans la liste déroulante afin de déterminer le corps du message à écrire.
Récupérer le message

Procédure
1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** pour exécuter le Job.

Récupérer le premier message de la file de messages

Créez ensuite le second Job pour récupérer le premier message de la file d’attente.

```
Starting job MQInput at 16:57 18/11/2010
[statistics] connecting to socket on port 3719
[statistics] connected
OpenQueueWithAccess
(DIRECT=OS:localhost\private$\AddressQueue)
open:
    fsname(DIRECT=OS:localhost\private$\AddressQueue)
    accessmode(1) sharemode(0)
open:
    fsname(DIRECT=OS:localhost\private$\AddressQueue)
    accessmode(2) sharemode(0)
Atlanta
[statistics] disconnected
Job MQInput ended at 16:57 18/11/2010 [exit code=0]
```

Le corps du message *Atlanta* récupéré de la file d’attente s’affiche dans la console.

Construire le flux du message

Procédure
1. Déposez un **tMicrosoftMQInput** et un **tLogRow** de la **Palette** dans l’espace de modélisation graphique.
2. Reliez ces deux composants à l’aide d’un lien **Row > Main**.

Configurer le traitement du message

Procédure
1. Double-cliquez sur le **tMicrosoftMQInput** afin d’ouvrir sa vue **Component**.
2. Dans le champ **Host**, saisissez le nom ou l'adresse de l'hôte. Ici, saisissez *localhost*.

3. Dans le champ **Queue**, saisissez le nom de la file d'attente de message à partir de laquelle vous souhaitez récupérer le message. Dans cet exemple, saisissez *AddressQueue*.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour exécuter le Job.
**tMicrosoftMQOutput**

Ce composant écrit une colonne définie d’un flux d’entrée de données dans une file d’attente de messages Microsoft (le composant supporte uniquement le type String).

**Propriétés du tMicrosoftMQOutput Standard**

Ces propriétés sont utilisées pour configurer le tMicrosoftMQOutput s’exécutant dans le framework de Jobs Standard.

Le composant tMicrosoftMQOutput Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PROPERTY</strong></td>
<td>Peut être <em>Built-in</em> ou <em>Repository.</em></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte.</td>
</tr>
<tr>
<td><strong>Queue</strong></td>
<td>Saisissez le nom de la file de messages dans laquelle vous souhaitez écrire un message. Cette file peut être créée automatiquement à la volée si elle n’existe pas.</td>
</tr>
</tbody>
</table>
| **Queue type** | Sélectionnez un type de file d’attente de message Microsoft dans la liste déroulante.  
- **Private Non-Transaction** : une queue privée non-transactionnelle.  
- **Private Transaction** : une queue privée transactionnelle.  
- **Public Non-Transaction** : une queue publique non-transactionnelle.  
- **Public Transaction** : une queue publique transactionnelle.  
| **Message column** | Sélectionnez le message à écrire dans la file de messages Microsoft. Le message sélectionné doit être de type String. |
### Die on error

Cette case est décochée par défaut, pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

### Advanced settings

| tStatCatcher Statistics | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

### Global Variables


### Utilisation

| Règle d’utilisation | Ce composant doit être lié à un composant d’entrée ou intermédiaire. |

| Connexions | Liens de sortie (de ce composant à un autre) :

**Row** : Main, Iterate.

**Trigger** : Run if, On Component Ok, On Component Error.

Liens d’entrée (d’un autre composant à celui-ci) :

**Row** : Main, Reject, Iterate


Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend. |

| Limitation | Le message à écrire ne peut être null.

Ce composant requiert l’installation des fichiers .jar liés. |
Scénario associé

Pour un scénario associé, consultez Scénario : Ecrire et récupérer des messages dans une file d’attente de message Microsoft à la page 2396.
tMomCommit

Ce composant commite des données sur le serveur MQ.
Le tMomCommit utilise une connexion unique pour commiter en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes et permet donc un gain de performance.

**Propriétés du tMomCommit Standard**

Ces propriétés sont utilisées pour configurer le tMomCommit s’exécutant dans le framework de Jobs Standard.
Le composant tMomCommit Standard appartient à la famille Internet.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Component List</th>
<th>Sélectionnez dans la liste le composant de connexion du le Job en cours.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Avertissement :**
Si vous utilisez un lien de type Row > Main pour relier le tMomCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace |
La page de document contient des informations sur le composant tMomCommit. Voici la traduction des informations extraites:

**pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.**

**Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.**

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants Mom, notamment avec le tMomConnection et le tMomRollback.</th>
</tr>
</thead>
</table>

### Scénario associé

Pour plus d’informations relatives au fonctionnement du composant tMomCommit, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
**tMomConnection**

Ce composant ouvre une connexion vers un serveur MQ dans un but de communication.

**Propriétés du tMomConnection Standard**

Ces propriétés sont utilisées pour configurer le tMomConnection s’exécutant dans le framework de Jobs Standard.

Le composant tMomConnection Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failover</td>
<td>Cochez cette case pour utiliser le transport pour le failover ActiveMQ. Pour plus d’informations concernant le transport relatif au failover, consultez Failover Transport Reference (en anglais). Cette case est disponible lorsque vous sélectionnez ActiveMQ dans la liste MQ Server et que la case Static Discovery est décochée.</td>
</tr>
<tr>
<td>Static Discovery</td>
<td>Cochez cette case pour utiliser le mécanisme de transport statique ActiveMQ. Pour plus d’informations concernant le transport statique, consultez Static Transport Reference (en anglais). Cette case est disponible lorsque vous sélectionnez ActiveMQ dans la liste MQ Server et que la case Static Discovery est décochée.</td>
</tr>
<tr>
<td>URI parameters</td>
<td>Saisissez les paramètres de l’URI. Ce champ est disponible lorsque la case Failover ou Static Discovery est cochée.</td>
</tr>
<tr>
<td>MQ server</td>
<td>Sélectionnez dans la liste déroulante le type des serveurs MQ à connecter, ActiveMQ ou WebSphere MQ.</td>
</tr>
<tr>
<td>Use SSL Transport</td>
<td>Cochez cette case pour utiliser le transport ActiveMQ SSL. Pour plus d’informations concernant le transport SSL, consultez SSL Transport Reference (en anglais). Cette case est disponible uniquement lorsque vous sélectionnez ActiveMQ dans la liste MQ server.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP ou nom de l’hébergeur du serveur MQ.</td>
</tr>
<tr>
<td>Port</td>
<td>Port d’écoute du serveur MQ.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets</td>
</tr>
</tbody>
</table>
doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Use Transacted**
Cochez cette case pour configurer la session pour laquelle effectuer une transaction.
Cette case est disponible si vous sélectionnez ActiveMQ dans la liste déroulante MQ server.

**Channel**
Spécifiez le nom du canal dans lequel la connexion est établie. La valeur par défaut est DC.SVRCONN.
Ce champ est disponible si vous sélectionnez WebSphere MQ dans la liste déroulante du serveur MQ server.

**QueueManager**
Spécifiez le nom du gestionnaire de file auquel la connexion est établie.
Ce champ est disponible si vous sélectionnez WebSphere MQ dans la liste déroulante du serveur MQ server.

**Use or register a shared Connection**
Cochez cette case pour partager votre connexion ou récupérer une connexion mise en partage dans un Job père ou un Job fils. Cette option vous permet de partager une seule connexion à une base de données dans plusieurs composants de connexion à des bases de données, dans différents niveaux de Jobs, qui peuvent être des Jobs parent ou enfant.
Cette option est incompatible avec les options Use dynamic job et Use an independent process to run subjob du composant tRunJob. Utiliser une connexion partagée avec un tRunJob ayant l’une de ces options activée fera échouer votre Job.

**Shared Connection Name**
Saisissez le nom de la connexion partagée.
Ce champ est disponible si vous cochez la case Use or register a shared Connection.

**Advanced settings**

**Additional options**
Définissez dans la table les options de transport supplémentaires. Cette table est disponible uniquement si vous avez coché la case Failover ou Static Discovery dans la vue Basic settings.
- **Option name** : nom de l’option.
- **Option value** : valeur de l’option.

Par exemple, saisissez maxReconnectAttempts dans le champ Option name puis 0 dans le champ Option value afin qu’aucune tentative ne soit effectuée pour récupérer la connexion lorsqu’aucun serveur n’est disponible.

Pour plus d’informations concernant les options de transport liées au failover, consultez Failover Transport Reference (en anglais).
Pour plus d’informations concernant les options de transport statique, consultez Static Transport Reference (en anglais).

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec des composants Mom, notamment les composants **tMomCommit** et **tMomRollback**. |

| Limitation | Ce composant supporte les serveurs ActiveMQ et WebSphereMQ. |

### Scénario associé

Pour un scénario associé, consultez la **tMysqlConnection** à la page 2618.
tMomInput

Ce composant récupère un message d’une file d’attente d’un système Middleware orienté Message (MOM) et le passe au composant suivant.
Le tMomInput permet de mettre en place des communications asynchrones via un serveur MOM.

**Propriétés du tMomInput Standard**

Ces propriétés sont utilisées pour configurer le tMomInput s’exécutant dans le framework de Jobs Standard.
Le composant tMomInput Standard appartient à la famille Internet.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.</td>
</tr>
<tr>
<td>Keep Listening</td>
<td>Cochez cette case pour consulter en continu le serveur MOM pour récupérer les nouveaux messages. Cette case n’est pas disponible lorsque vous sélectionnez ActiveMQ, dans la liste MQ server et que la case receive number of messages est cochée.</td>
</tr>
</tbody>
</table>
| Sleeping time (in sec)             | Saisissez en secondes l’intervalle de temps durant lequel le message suivant doit être reçu. Ce champ est disponible dans les situations suivantes :  
- JBoss Messaging est sélectionné dans la liste MQ server et la case Keep Listening est cochée.  
- ActiveMQ est sélectionné dans la liste MQ server et la case Keep Listening est cochée.  
- ActiveMQ est sélectionné dans la liste MQ server et les cases receive number of messages et Timeout for receiver sont cochées. |
<p>| receive number of messages          | Cochez cette case pour configurer le nombre maximal de messages pouvant être reçus. Cette case est disponible uniquement lorsque vous sélectionnez ActiveMQ dans la liste MQ server et que la case Keep Listening est cochée. |
| maximum messages                   | Saisissez le nombre maximal de messages pouvant être reçus.                                                                                                                                                 |</p>
<table>
<thead>
<tr>
<th><strong>MomInput</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ce champ est disponible lorsque vous cochez la case receive number of messages.</strong></td>
<td></td>
</tr>
</tbody>
</table>
| **Timeout for receiver** | Cochez cette case et spécifiez l’intervalle de temps avant suspension de la réception, dans le champ **Sleeping time (in sec).**  
Cette case est disponible lorsque la case **receive number of messages** est cochée. |
| **Failover** | Cochez cette case pour utiliser le transport pour le failover ActiveMQ. Pour plus d’informations concernant le transport relatif au failover, consultez **Failover Transport Reference** (en anglais).  
Cette case est disponible lorsque vous sélectionnez ActiveMQ dans la liste **MQ Server** et que la case **Static Discovery** est décochée. |
| **Static Discovery** | Cochez cette case pour utiliser le mécanisme de transport statique ActiveMQ. Pour plus d’informations concernant le transport statique, consultez **Static Transport Reference** (en anglais).  
Cette case est disponible lorsque vous sélectionnez ActiveMQ dans la liste **MQ Server** et que la case **Static Discovery** est décochée. |
| **URI parameters** | Saisissez les paramètres de l’URI.  
Ce champ est disponible lorsque la case **Failover** ou **Static Discovery** est cochée. |
| **MQ server** | Sélectionnez dans la liste le type de serveur MQ auquel se connecter : ActiveMQ, JBoss Messaging, ou WebSphere MQ. En fonction du serveur sélectionné, les paramètres diffèrent légèrement. |
| **Use SSL Transport** | Cochez cette case pour utiliser le transport ActiveMQ SSL. Pour plus d’informations concernant le transport SSL, consultez **SSL Transport Reference** (en anglais).  
Cette case est disponible uniquement lorsque vous sélectionnez ActiveMQ dans la liste **MQ server.** |
| **Host** | Renseignez le nom de l’hôte ou l’adresse IP du serveur MQ. |
| **Port** | Saisissez le numéro du port d’écoute du serveur MQ. |
| **Username et Password** | Identifiant de connexion au serveur.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Channel** | Spécifiez le nom du canal via lequel la connexion est établie. La valeur par défaut est **DC.SVRCONN.** |
| **Message From** | Saisissez la source du message, telle qu’attendue par le serveur. Celle-ci peut être composé du type et du nom de la source, par exemple : queue/A ou topic/testtopic. Notez que ce champ est sensible à la casse. Ce champ est disponible si vous sélectionnez ActiveMQ ou JBoss Messaging dans la liste MQ Server. |
| **Message Type** | Sélectionnez le type de message : topic ou queue. Cette liste est disponible si vous sélectionnez ActiveMQ ou JBoss Messaging dans la liste MQ Server. |
| **Message Body Type** | Sélectionnez le type de message : Text Message, Bytes Message ou Map Message. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma du composant tMomInput est composé de deux colonnes : From et Message, dont le nom est en lecture seule. |
| **QueueManager** | Spécifiez le nom du gestionnaire des files d’attente auquel la connexion est établie. Ce champ est disponible uniquement lorsque vous sélectionnez WebSphere MQ dans la liste MQ server. |
| **MessageQueue** | Saisissez le nom de la file d’attente des messages dans laquelle les applications de mise en file d’attente mettent les messages. Ce champ est disponible uniquement lorsque vous sélectionnez WebSphere MQ dans la liste MQ server. |
| **Is using message id to fetch** | Cochez cette case pour appliquer un filtre portant sur les messages id, filtre que vous déterminerez dans le champ Message id. Cette case est disponible lorsque vous sélectionnez Text Message ou Byte Message dans la liste Message Body Type. |
| **Commit (delete message after read from the queue)** | Cochez cette case afin de forcer le commit après lecture de chaque message dans la file d’attente. Ce champ est disponible uniquement lorsque vous sélectionnez WebSphere MQ dans la liste MQ server et que la case Browse Message est décochée dans la vue Advanced settings. |
| **Backout messages to input queue** | Cochez cette case pour remettre dans la file le message après l’avoir lu. Cette case et la case Browse Message de la vue Advanced settings vous permettent de lire des messages depuis la file sans les détruire. Pour plus d’informations |

Cette case est disponible uniquement lorsque vous sélectionnez **WebSphere MQ** dans la liste **MQ server** et que la case **Keep listening** est décochée. Pour plus d'informations, consultez le [Guide MQSeries Using Java](https://publib.boulder.ibm.com/iseries/v5r2/ic2924/books/csqzaw07.pdf) (en anglais).

**Backout messages to backout queue (when backoutCount of message reach the threshold value)**

Cochez cette case pour effectuer un backout sur les messages dans la file de backout définie dans le gestionnaire WebSphere MQ lorsque le nombre de backout atteint le seuil configuré, qui est une propriété de la file du serveur Websphere MQ.

Cette case est disponible lorsque vous sélectionnez **WebSphere MQ** dans la liste **MQ server** et que les cases **Keep Listening**, **Commit (delete message after read from the queue)** et **Browse message** sont décochées.

**Set Transacted**


**Remarque :**

Cochez cette case pour ignorer les paramètres de la liste **Acknowledgement Mode** dans la vue **Advanced settings** du tMomInput.

Cette case est disponible uniquement lorsque vous sélectionnez **ActiveMQ** dans la liste **MQ Server**.

**start server**

Cochez cette case pour forcer le démarrage du serveur ActiveMQ.

Cette case est disponible uniquement lorsque vous sélectionnez **ActiveMQ** dans la liste **MQ server**.

**Advanced settings**

**Additional options**

Définissez dans la table les options de transport supplémentaires. Cette table est disponible uniquement si vous avez coché la case **Failover** ou **Static Discovery** dans la vue **Basic settings**.

- **Option name** : nom de l'option.
- **Option value** : valeur de l'option.

Par exemple, saisissez `maxReconnectAttempts` dans le champ **Option name**, puis `0` dans le champ **Option value** afin qu'aucune tentative ne soit effectuée pour récupérer la connexion lorsqu'aucun serveur n’est disponible.

Pour plus d’informations concernant les options de transport liées au failover, consultez [Failover Transport Reference](https://publib.boulder.ibm.com/iseries/v5r2/ic2924/books/csqzaw07.pdf) (en anglais).

Pour plus d’informations concernant les options de transport statique, consultez [Static Transport Reference](https://publib.boulder.ibm.com/iseries/v5r2/ic2924/books/csqzaw07.pdf) (en anglais).
### Acknowledgement Mode

Sélectionnez dans la liste un mode d’accusé de réception pour indiquer que le client a bien reçu le(s) message(s) :

- **Auto Acknowledge** : avec ce mode d’accusé de réception, l’acquittement se fait automatiquement, le client reçoit l’accusé de réception à la réception du message, qu’il soit traité ou non par l’application.

- **Client Acknowledge** : avec ce mode d’accusé réception, le client accuse réception d’un message en appelant la méthode d’accusé réception du message.

- **Dups OK Acknowledge** : ce mode d’accusé réception permet d’indiquer au client que le message peut être envoyé plusieurs fois à une destination. Ce mode permet d’améliorer les performances de certains clients, lorsque les messages sont nombreux.


**Remarque :**
Si la case **Set Transacted** du **tMomOutput** est cochée, l’option **Acknowledgement Mode** est ignorée. Cette case est activée lorsque le serveur MQ est **ActiveMQ** ou **JBoss Messaging**.

### Get Jms Header

Cochez cette case pour recevoir les en-têtes JMS via le mapping des champs JMS vers les champs MQ Series sur le serveur MQ. Lorsque cette case est cochée, vous pouvez spécifier le nom de l’en-tête JMS ainsi que le nom de la colonne de référence correspondante dans la (les) ligne(s) ajoutée(s) en cliquant sur le bouton "+" sous la table **Parameters**. Pour plus d’informations concernant l’utilisation des en-têtes JMS, consultez le Chapitre 12 de MQSeries Using Java (en anglais).

Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.

### Get Jms Properties

Cochez cette case pour recevoir les propriétés JMS mappées dans les champs MQMD. Lorsque cette case est cochée, vous pouvez spécifier le nom, le type et le nom de la colonne de référence dans la (les) ligne(s) ajoutée(s) en cliquant sur le bouton "+" sous la table **Parameters**. Pour plus d’informations concernant l’utilisation des propriétés JMS, consultez le Chapitre 12 de MQSeries Using Java.

Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.

### Browse message

Cochez cette case afin de désactiver les options **Commit (delete message after read from the queue)** et **Backout messages to input queue** dans la vue **Basic settings** et d’ouvrir la file pour parcourir les messages.

**Remarque :**
Cette case ainsi que la case **Backout messages to input queue** dans la vue **Basic settings** vous
permettent de lire des messages de manière non destructive dans la file. La case **Browse message** est visible uniquement lorsque la case **Backout messages to input queue** est décochée. Pour plus d’informations, consultez **MQC: MQOO_BROWSE** (en anglais).

### Get MQMD Fields

Cochez cette case pour définir un ou plusieurs Message Descriptor(s) en ajoutant de nouveaux champs pour MQMD (Message queuing message descriptor) dans le tableau **Parameter** :

**Field Name** : sélectionnez un ou plusieurs descripteur(s) de messages dans la liste afin de récupérer les informations de l’en-tête du message.

**Reference Column Name** : Informations d’en-tête et propriétés du message.

Pour plus d’informations, consultez **Fields for MQMD** (en anglais).

Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.

### Include Header

Cochez cette case pour définir l’en-tête du message :

**MQRFH2 fixed Portion** : Cochez cette case et cliquez sur le bouton [+] pour ajouter une ou plusieurs ligne(s) pour spécifier la **Version**, l’encode (**Encoding**), le **Format**, les **Flags** et les valeurs **NameValueCCSID** et **CodeCharSetId** pour la portion fixe de l’en-tête MQRFH2.

**MCD folder** : Cochez cette case et cliquez sur le bouton [+] pour ajouter une ou plusieurs ligne(s) pour spécifier le nom et la valeur des propriétés décrivant le format du message.

**JMS folder** : Cochez cette case et cliquez sur le bouton [+] pour ajouter une ou plusieurs ligne(s) pour spécifier le nom du champ et sa valeur pour le transport des champs d’en-têtes JMS et des propriétés JMSX.

**USR folder** : Cochez cette case et cliquez sur le bouton [+] pour ajouter une ou plusieurs ligne(s) pour spécifier le nom du champ et sa valeur pour le transport des propriétés définies par l’application associée au message.


Ce champ est disponible si vous sélectionnez **ActiveMQ** ou **JBoss Messaging** dans la liste **MQ Server**.

### Set CipherSpec

Cochez cette case pour activer la liste **CipherSpec** dans laquelle vous pouvez spécifier le CipherSpec à utiliser avec le SSL WebSphere.

Pour plus d’informations concernant CipherSpec, consultez **Specifying CipherSpecs** (en anglais).
Ce champ est disponible si vous sélectionnez ActiveMQ ou JBoss Messaging dans la liste MQ Server.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

- **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le [Guide utilisateur du Studio Talend](#).

### Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé en tant que composant de début. Il requiert un composant de sortie.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion au serveur MQ parmi celles prévues dans votre Job.

La table *Dynamic settings* est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue *Basic settings*. Lorsqu’un paramètre dynamique est configuré, la liste *Component List* de la vue *Basic settings* devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez [Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte](#) à la page 2641 et [Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement](#) à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le [Guide utilisateur du Studio Talend](#).
**Scénario 1 : Communication asynchrone via un serveur MOM**

Ce scénario est composé de deux Jobs. Le premier permet de poster des messages sur la file d’attente du serveur JBoss et le deuxième récupère les messages de ce serveur.

**Construire et exécuter le premier Job**

**Pourquoi et quand exécuter cette tâche**

Dans le premier Job, un message composé de chaînes de caractères est créé à l’aide du composant `tRowGenerator` puis transmis au serveur JBoss à l’aide du composant `tMomOutput`. Un composant intermédiaire, le composant `tLogRow`, permet d’afficher le flux transmis.

**Procédure**

1. Cliquez et déposez ces trois composants dans votre premier Job et connectez-les via le clic-droit avec des liens de type **Main > Row**.
2. Double-cliquez sur le composant `tRowGenerator` pour configurer le schéma des données qui seront générées de manière aléatoire.

3. Créez une seule colonne appelée `message`. Elle correspond au message qui sera mis sur la file d’attente du MOM.
5. Paramétrez le champ **Number of rows to be generated** permettant de définir le nombre de lignes générée à 10.
   Cliquez sur **OK** pour valider.

6. Le composant **tLogRow** est uniquement utilisé pour afficher un état intermédiaire des données manipulées. Dans cet exemple, aucune configuration spécifique n’est nécessaire.

7. Sélectionnez le composant **tMomOutput**.

<table>
<thead>
<tr>
<th>MQ server</th>
<th>Host</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBoss</td>
<td>localhost</td>
<td>1099</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TO</th>
<th>Message Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>queue/A</td>
<td>Queue</td>
</tr>
</tbody>
</table>

8. Dans cet exemple, le serveur **MQ server** à utiliser est de type **JBoss**.
   Dans les champs **Host** et **Port**, renseignez les informations de connexion correspondantes.

9. Sélectionnez le type de message dans la liste déroulante **Message type**, il peut être de type **Queue** ou **Topic**. Pour cet exemple, sélectionnez le type **Queue** dans la liste.

10. Dans le champ **To**, saisissez les informations source du message strictement telles qu’elles sont attendues par le serveur. Il doit correspondre au type de message que vous avez sélectionné dans la liste déroulante **Message Type**, notamment : **queue/A**.

   **Remarque :**
   Le nom du message est sensible à la casse, ainsi **queue/A** et **Queue/A** sont différents.

11. Puis cliquez sur **Sync Columns** pour récupérer le schéma du composant précédent. Le schéma étant en lecture seule, il ne peut être modifié. Les données envoyées sur le serveur MQ proviennent de la première colonne rencontrée du schéma.

12. Exécutez le Job en appuyant sur **F6** et, grâce au composant **tLogRow**, observez dans la console le flux de données transmis.

**Résultats**

```
c8X5C0
1EhC41
dpGyQP
x6gQt8
E3knZp
tXU2E?
6H7lv1
SU8eM
2wewv@
xfPZAP
Job tMomInput ended at 17:46 14/09/2007. [exit code=0]
```

**Construire et exécuter le second Job**

**Pourquoi et quand exécuter cette tâche**

Puis paramétrez le deuxième Job pour récupérer les messages de la file d’attente du serveur MOM.
Procédure

1. Cliquez et déposez un composant tMomInput (dans la famille Internet de la Palette) et un composant tLogRow pour afficher les messages récupérés.

2. Sélectionnez le tMomInput et définissez ses paramètres.

   - Keep Listening : cochez la case
   - Sleeping Time (in sec) : 5
   - MQ server : JBoss Messaging
   - Host : 10.42.10.96
   - Port : 1099
   - Message From : queue/A
   - Message Type : Queue
   - Schema : lane


4. Renseignez les informations Host et Port du serveur.

5. Renseignez les paramètres des champs Message From et Message Type afin qu’ils correspondent à la source et au type de message attendus par le serveur de message.


7. Cochez la case Keep listening et paramétrez la fréquence de vérification à 5 secondes.

   **Remarque :**

   Lorsque vous utilisez l’option Keep Listening, vous devez cliquer sur le bouton Kill pour stopper le Job.

8. Laissez les paramètres par défaut du composant tLogRow.

9. Enregistrez le Job et exécutez-le (si vous le lancez pour la première fois ou si vous l’avez stoppé au cours d’un lancement précédent).

Résultats

Starting job momoutput at 17:47 14/09/2007
[statistics] connecting to socket on port 3414
[statistics] connected
Ready to receive message
Waiting...
queue/A|c8X6GC
queue/A|1EhC41
queue/A|OpGyQP
queue/A|x6g0t8
queue/A|ESkuZp
queue/A|tXU2ET
queue/A|6H7Nwl
queue/A|8UA9eM
queue/A|2sabwV
queue/A|tFZAP
Job momoutput ended at 17:47 14/09/2007. [exit code=1]

Les messages récupérés sur le serveur sont affichés dans la console.
**Scénario 2 : Transmettre des fichiers XML via un serveur MOM**

Ce scénario décrit un Job à cinq composants, comprenant deux sous-jobs envoyant des fichiers XML depuis un dossier local vers une file MOM puis récupérant les fichiers de la file MOM et affichant le contenu des fichiers dans la console.

**Déposer et relier les composants**

**Procédure**

1. De la Palette, déposez les composants suivants dans l'espace de modélisation graphique : un tFileList, un tFileInputXML, un tMomOutput, un tMomInput et un tLogRow.
2. Reliez le tFileList au tFileInputXML à l'aide d'un lien Row > Iterate et reliez le tFileInputXML au tMomOutput à l'aide d'un lien Row > Main afin de former le premier sous-job. Ce sous-job lit chaque fichier XML dans un dossier donné et l'envoie dans une file MOM.
3. Connectez le tMomInput au tLogRow à l'aide d'un lien Row > Main pour former le second sous-job. Ce sous-job récupère les fichiers XML de la file MOM et affiche leur contenu dans la console.
4. Reliez le tFileInputXML au tMomInput à l'aide d'un lien Trigger > On Component Ok afin de relier les deux sous-jobs.

**Configurer le premier sous-job**

**Configurer les composants d'entrée**

**Procédure**

1. Double-cliquez sur le tFileList pour ouvrir sa vue Basic settings.
2. Dans le champ Directory, saisissez le chemin d'accès au répertoire duquel lire les fichiers XML, ou parcourez votre système en cliquant sur le bouton [...].

3. Cochez la case Use Glob Expressions as Filemask, ajoutez une nouvelle ligne dans la table Files en cliquant sur le bouton [+] et saisissez "*.xml" comme masque de fichier, dans la colonne Filemask, afin d'utiliser tous les fichiers XML du répertoire. Laissez les autres paramètres tels qu'ils sont.

4. Double-cliquez sur le composant tFileInputXML pour ouvrir sa vue Basic settings.

5. Cliquez sur le bouton [...] à côté du champ Edit schema pour ouvrir la boîte de dialogue [Schema].
6. Cliquez sur le bouton [+] pour ajouter une colonne, nommez-la *message* dans cet exemple et sélectionnez **Document** dans la liste **Type** afin de gérer les fichiers au format XML. Cliquez sur **OK** pour fermer la boîte de dialogue.

7. Dans le champ **File name/Stream**, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables globales et sélectionnez `tFileList_1.CURRENT_FILEPATH` afin d’effectuer une boucle sur le répertoire des fichiers de contexte.

8. Dans le champ **Loop XPath query**, saisissez `/` pour définir la racine comme nœud de boucle dans la structure des fichiers d’entrée. Dans la table **Mapping**, dans la colonne **XPath query**, saisissez `./` pour extraire toutes les données du nœud de contexte des fichiers source et cochez la case **Get Nodes** afin de construire un flux de données de type **Document**.

**Configurer le composant tMomOutput**

**Procédure**

1. Double-cliquez sur le composant **tMomOutput** pour ouvrir sa vue **Basic settings**.

2. Sélectionnez **WebSphere MQ** dans la liste **MQ server** et saisissez le nom de l’hôte ou l’adresse IP du serveur MQ, ainsi que le numéro du port.

3. Saisissez les informations d’authentification dans les champs **Username** (identifiant) et **Password** (mot de passe) puis saisissez le nom du canal de la file de transmission, dans le champ **Channel**.
4. Comme vous gérez des messages de fichiers, sélectionnez **Text Message** dans la liste **Message Body Type**.

5. Cliquez sur **Sync columns** afin de récupérer la structure du schéma du composant précédent.

6. Renseignez les informations relatives au gestionnaire de file, dans le champ **QueueManager** et les informations concernant la file de messages, dans le champ **MessageQueue**. Laissez les autres paramètres tels qu’ils sont.

**Configurer le second sous-job**

**Procédure**

1. Double-cliquez sur le **tMomInput** pour ouvrir sa vue **Basic settings**.

2. Configurez les paramètres simples du composant de la même manière que pour le **tMomOutput**, y compris les informations du serveur MQ, les informations d’authentification, le canal, le type de message, le gestionnaire de file et la file de messages.

3. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir la boîte de dialogue **[Schema]**.

4. Dans la liste **Type** de la colonne **message**, sélectionnez **Document** afin de gérer les formats XML puis cliquez sur **OK** pour fermer la boîte de dialogue.
Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur les touches Ctrl+S afin de sauvegarder votre Job.
2. Appuyez sur F6 ou cliquez sur Run dans l'onglet Run pour exécuter le Job.

Les fichiers XML du dossier spécifié sont écrits dans la file de messages puis sont récupérés de la file. Le contenu des fichiers est affiché dans la console.
tMomMessageIdList

Ce composant récupère la liste des ID des messages à partir de la file d’attente d’un système Middleware orienté Message et la passe au composant suivant.

Le tMomMessageIdList itère sur les ID de certains messages. Il est généralement utilisé avec le tMomInput. Pour plus d’informations, consultez Propriétés du tMomInput Standard à la page 2409.

Propriétés du tMomMessageIdList Standard

Ces propriétés sont utilisées pour configurer le tMomMessageIdList s’exécutant dans le framework de Jobs Standard.

Le composant tMomMessageIdList Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

| Use existing connection | Cochez cette case et, dans la liste qui s’affiche, sélectionnez le composant de connexion permettant de réutiliser les informations de connexion précédemment définies.
Lorsqu’un Job contient un Job parent et un Job enfant, la liste qui s’affiche présente uniquement les composants de connexion du Job du même niveau. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MQ Server</td>
<td>Sélectionnez dans cette liste le serveur MOM à utiliser. Les paramètres requis diffèrent légèrement en fonction du serveur sélectionné.</td>
</tr>
<tr>
<td>Host/Port</td>
<td>Renseignez le nom de l’hôte (Host) et l’adresse IP du serveur MOM ainsi que le Port.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option. CURRENT_MESSAGE_ID : l’identifiant du message courant. Cette variable est une variable Flow et retourne une chaîne de caractères. NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en tant que composant de début. Il requiert un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Websphere</strong></td>
<td><strong>Channel</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Queue Manager</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Message Queue</strong></td>
</tr>
</tbody>
</table>

| Limitation          | Assurez-vous que le serveur Websphere correspondant est lancé.                                  |
|                     | Ce composant requiert l’installation des fichiers .jar liés.                                    |

### Scénario associé

Pour un scénario associé, consultez **tMomInput** à la page 2409.
**tMomOutput**

Ce composant ajoute un message dans la file d’attente d’un système Middleware orienté message afin qu’il soit récupéré de manière asynchrone.

Le tMomOutput permet de mettre en place des communications asynchrones via un serveur MOM.

**Propriétés du tMomOutput Standard**

Ces propriétés sont utilisées pour configurer le tMomOutput s’exécutant dans le framework de Jobs Standard.

Le composant tMomOutput Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Use existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failover</td>
<td>Cochez cette case pour utiliser le transport pour le failover ActiveMQ. Pour plus d’informations concernant le transport relatif au failover, consultez Failover Transport Reference (en anglais). Cette case est disponible lorsque vous sélectionnez ActiveMQ dans la liste MQ Server et que la case Static Discovery est décochée.</td>
</tr>
<tr>
<td>Static Discovery</td>
<td>Cochez cette case pour utiliser le mécanisme de transport statique ActiveMQ. Pour plus d’informations concernant le transport statique, consultez Static Transport Reference (en anglais). Cette case est disponible lorsque vous sélectionnez ActiveMQ dans la liste MQ Server et que la case Static Discovery est décochée.</td>
</tr>
<tr>
<td>URI parameters</td>
<td>Saisissez les paramètres de l’URI. Ce champ est disponible uniquement lorsque la case Failover ou Static Discovery est cochée.</td>
</tr>
<tr>
<td>MQ server</td>
<td>Sélectionnez dans la liste le type de serveur MQ auquel se connecter : ActiveMQ, JBoss Messaging, ou WebSphere MQ. En fonction du serveur sélectionné, les paramètres diffèrent légèrement.</td>
</tr>
<tr>
<td>Use SSL Transport</td>
<td>Cochez cette case pour utiliser le transport ActiveMQ SSL. Pour plus d’informations concernant le transport SSL, consultez SSL Transport Reference (en anglais).</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Renseignez le nom de l'hôte ou l'adresse IP du serveur MQ.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Saisissez le numéro du port d'écoute du serveur MQ.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Identifiant de connexion au serveur. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Channel</strong></td>
<td>Spécifiez le nom du canal via lequel la connexion est établie. La valeur par défaut est DC.SVRCONN. Ce champs est disponible uniquement lorsque vous sélectionnez WebSphere MQ dans la liste MQ server.</td>
</tr>
<tr>
<td><strong>To</strong></td>
<td>Saisissez la destination du message, telle qu’attendue par le serveur ; le type est le nom du dossier cible doivent être indiqués, par exemple : queue/A ou topic/testtopic Notez que ce champ est sensible à la casse. Ce champ est disponible lorsque vous sélectionnez ActiveMQ ou JBoss Messaging dans la liste MQ Server.</td>
</tr>
<tr>
<td><strong>Message Type</strong></td>
<td>Sélectionnez le type de message : topic ou queue. Cette liste est disponible lorsque vous sélectionnez ActiveMQ ou JBoss Messaging dans la liste MQ Server.</td>
</tr>
<tr>
<td><strong>Message Body Type</strong></td>
<td>Sélectionnez le type de message : Text Message, Bytes Message ou Map Message.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Lors de l’utilisation du composant tMomOutput, le schéma est en lecture seule mais changera en fonction du schéma d’entrée. Le serveur attend uniquement un schéma à une colonne pour contenir les Messages.</td>
</tr>
<tr>
<td><strong>QueueManager</strong></td>
<td>Spécifiez le nom du gestionnaire des files d’attente auquel la connexion est établie. Ce champ est disponible uniquement lorsque vous sélectionnez WebSphere MQ dans la liste MQ server.</td>
</tr>
<tr>
<td><strong>MessageQueue</strong></td>
<td>Saisissez le nom de la file d’attente des messages dans laquelle les applications de mise en file d’attente mettent les messages. Ce champ est disponible uniquement lorsque vous sélectionnez WebSphere MQ dans la liste MQ server.</td>
</tr>
<tr>
<td>Set Transacted</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Is using message id to set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour paramétrer les messages selon leur ID. Cette case est disponible uniquement lorsque vous sélectionnez Text Message ou Bytes Message dans la liste Message Body Type.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Additional options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Définissez dans la table les options de transport supplémentaires. Cette table est disponible uniquement si vous avez coché la case Failover ou Static Discovery dans la vue Basic settings.</td>
</tr>
<tr>
<td>• Option name : nom de l’option.</td>
</tr>
<tr>
<td>• Option value : valeur de l’option.</td>
</tr>
<tr>
<td>Par exemple, saisissez maxReconnectAttempts dans le champ Option name puis 0 dans le champ Option value afin qu’aucune tentative ne soit effectuée pour récupérer la connexion lorsqu’aucun serveur n’est disponible.</td>
</tr>
<tr>
<td>Pour plus d’informations concernant les options de transport liées au failover, consultez Failover Transport Reference (en anglais).</td>
</tr>
<tr>
<td>Pour plus d’informations concernant les options de transport statique, consultez Static Transport Reference (en anglais).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Delivery Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez un mode de livraison supporté par JMS :</td>
</tr>
<tr>
<td>Not Persistent : Ce mode de livraison ne nécessite pas que le message soit stocké de manière stable.</td>
</tr>
<tr>
<td>Persistent : Ce mode de livraison nécessite que le message soit stocké de manière stable, en tant que partie de l’opération d’envoi effectuée par le client.</td>
</tr>
<tr>
<td>Pour plus d’informations concernant les modes de livraison, consultez le Chapitre 14 de MQSeries Using Java (en anglais).</td>
</tr>
<tr>
<td>(Disponible lorsque vous sélectionnez ActiveMQ ou JBoss Messaging dans la liste MQ Server.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set Jms Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour envoyer les en-têtes JMS via le mapping des champs JMS vers les champs MQ Series sur le serveur MQ. Lorsque cette case est cochée, vous pouvez spécifier le nom de l’en-tête ainsi que sa valeur dans la (les) ligne(s) ajoutée(s) en cliquant sur le bouton [+] sous la table Parameters. Pour plus d’informations</td>
</tr>
</tbody>
</table>
concernant l’utilisation des en-têtes JMS, consultez le Chapitre 12 de MQSeries Using Java (en anglais).
Disponible lorsque vous sélectionnez ActiveMQ ou JBoss Messaging dans la liste MQ Server.

| Set Jms Properties | Cochez cette case pour envoyer les propriétés JMS mappées dans les champs MQMD du serveur MQ. Lorsque cette case est cochée, vous pouvez spécifier le nom, le type et la valeur de la propriété dans la (les) ligne(s) ajoutée(s) en cliquant sur le bouton [+] sous la table Parameters. Pour plus d’informations concernant l’utilisation des propriétés JMS, consultez le Chapitre 12 de MQSeries Using Java.
Disponible lorsque vous sélectionnez ActiveMQ ou JBoss Messaging dans la liste MQ Server. |
|-------------------|-------------------------------------------------------------------------------------------------|

| Use format | Cochez cette case pour spécifier le format de message WebSphere dans le champ WebSphere Message Format. Le format par défaut est MQSTR.
Pour plus d’informations concernant le format des messages WebSphere, consultez WebSphere Message Format (en anglais).
Disponible lorsque vous sélectionnez Websphere MQ dans la liste MQ Server. |
|-------------|-------------------------------------------------------------------------------------------------|

| Set MQMD Fields | Cochez cette case pour activer la table Parameters dans laquelle vous pouvez spécifier la valeur des champs MQMD.
Cette case est disponible uniquement lorsque vous sélectionnez Websphere MQ dans la liste MQ Server. |
|---------------|-------------------------------------------------------------------------------------------------|

| Parameters | Spécifiez la valeur des champs MQMD.
• **Field Name** : sélectionnez un champ MQMD dans la liste.
• **Field Value** : saisissez la valeur du champ MQMD correspondant.
Cette table est disponible uniquement lorsque la case Set MQMD Fields est cochée.
Pour plus d’informations concernant les champs MQMD, consultez Fields for MQMD (en anglais). |
|-------------|-------------------------------------------------------------------------------------------------|

| Include Header | Cochez cette case pour définir l’en-tête du message :
**MQRFH2 fixed Portion** : Cochez cette case et cliquez sur le bouton [+] pour ajouter une ou plusieurs ligne(s) pour spécifier la Version, l’encodage (Encoding), le Format, les Flags et les valeurs NameValueCCSID et CodeCharSetId pour la portion fixe de l’en-tête MQRFH2.
**MCD folder** : Cochez cette case et cliquez sur le bouton [+] pour ajouter une ou plusieurs ligne(s) pour spécifier le nom et la valeur des propriétés décrivant le format du message.
**JMS folder** : Cochez cette case et cliquez sur le bouton [+] pour ajouter une ou plusieurs ligne(s) pour spécifier |
|----------------|-------------------------------------------------------------------------------------------------|

2429
le nom du champ et sa valeur pour le transport des champs d'en-têtes JMS et des propriétés JMSX.

**USR folder** : Cochez cette case et cliquez sur le bouton [+] pour ajouter une ou plusieurs ligne(s) pour spécifier le nom du champ et sa valeur pour le transport des propriétés définies par l’application associée au message.


(Disponible lorsque vous sélectionnez **Websphere MQ** dans la liste **MQ Server**.)

---

### Set CipherSpec

Cochez cette case pour activer la liste **CipherSpec** dans laquelle vous pouvez spécifier le CipherSpec à utiliser avec le SSL WebSphere.

Pour plus d’informations concernant CipherSpec, consultez **Specifying CipherSpecs** (en anglais).

(Disponible lorsque vous sélectionnez **Websphere MQ** dans la liste **MQ Server**.)

---

### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

---

### Global Variables

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

---

### Utilisation

**Règle d’utilisation**

Ce composant nécessite un composant d’entrée ou un composant intermédiaire.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de
contexte afin de sélectionner dynamiquement votre connexion au serveur MQ parmi celles prévues dans votre Job.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Limitation**

Assurez-vous que le serveur Websphere, JBoss ou ActiveMQ correspondant est bien lancé.

Ce composant requiert l’installation des fichiers .jar liés.

**Scénario associé**

Pour un exemple d’utilisation du tMomOutput, consultez tMomInput à la page 2409.
tMomRollback

Ce composant annule la transaction commitée sur le serveur MQ.

Propriétés du tMomRollback Standard

Ces propriétés sont utilisées pour configurer le tMomRollback s’exécutant dans le framework de Jobs Standard.
Le composant tMomRollback Standard appartient à la famille Internet.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Component List</th>
<th>Sélectionnez dans la liste le composant utilisé dans le Job.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backout messages to backout queue (WebsphereMQ)</td>
<td>Cochez cette case afin d’effectuer un backout sur les messages dans la file de backout définie dans le gestionnaire WebSphere MQ, lorsque le nombre de backouts atteint le seuil configuré. Ce seuil est une propriété de la file d’attente du serveur WebSphere MQ.</td>
</tr>
</tbody>
</table>

- Si un composant tMomConnection est sélectionné dans la liste Component List, un backout est effectué sur les messages traités par les composants tMomInput réutilisant la connexion créée par le tMomConnection.

- Si un composant tMomInput est sélectionné dans la liste Component List et que la case Use existing connection est décochée dans la vue Basic settings du tMomInput, un backout est effectué sur les messages traités par le composant tMomInput sélectionné.

- Si un composant tMomInput est sélectionné dans la liste Component List et que la case Use existing connection est cochée dans la vue Basic settings du tMomInput, un backout est effectué sur les messages traités par les composants tMomInput réutilisant la connexion utilisée par le tMomInput sélectionné.

- Si un composant tMomOutput est sélectionné dans la liste Component List et que la case Use existing connection est décochée dans la vue Basic settings du tMomOutput, aucun backout n’est effectué sur les messages.

- Si un composant tMomOutput est sélectionné dans la liste Component List et que la case Use existing connection est cochée dans la vue Basic settings du tMomOutput, un backout est effectué sur les messages traités par les composants tMomInput réutilisant la connexion utilisée par le tMomOutput sélectionné.
Notez que cette case fonctionne uniquement lorsque les cases Commit (delete message after read from the queue), Backout messages to input queue, Backout messages to backout queue (when backoutCount of message reach the threshold value) et Browse message de tous les composants tMomInput du Job sont décochées.

<table>
<thead>
<tr>
<th>MQ Server</th>
<th>Sélectionnez dans cette liste le serveur MOM à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en association avec des composants Mom, notamment avec le tMomConnection et le tMomCommit. |

**Scénario associé**

Pour un scénario associé, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623 du composant tMysqlRollback.
tMondrianInput

Ce composant exécute une requête d’expression multi-dimensionnelle (MDX) correspondant à la structure du groupe de données et au schéma défini.

Le tMondrianInput lit des données à partir de bases relationnelles et produit des groupes de données multidimensionnels reposant sur une requête MDX. Puis il passe les données multidimensionnelles obtenues au composant suivant à l’aide d’un lien de type Row Main.

Propriétés du tMondrianInput Standard

Ces propriétés sont utilisées pour configurer le tMondrianInput s’exécutant dans le framework de Jobs Standard.

Le composant tMondrianInput Standard appartient à la famille Business Intelligence.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Mondrian Version</th>
<th>Sélectionnez la version de Mondrian que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB type</td>
<td>Sélectionnez le type de relation approprié pour relier les bases de données.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Datasource</td>
<td>Chemin d’accès et nom du fichier contenant les données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
  * View schema : sélectionnez cette option afin de voir le schéma. |
### Change to built-in property
- Sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

### Update repository connection
- Sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th>Built-in</th>
<th>Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</th>
</tr>
</thead>
</table>

**Catalog**
- Chemin d'accès au catalogue (structure de l'entrepôt de données).

**MDX Query**
- Saisissez votre requête MDX en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma et dans la structure de l'entrepôt de données.

**Encoding**
- Sélectionnez le type d'encodage à partir de la liste ou choisissez l'option Custom pour le définir manuellement. Ce champ doit obligatoirement être renseigné.

### Advanced settings

**tStatCatcher Statistics**
- Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>NB_LINE</strong> : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable Flow et retourne un entier. <strong>QUERY</strong> : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères. <strong>ERROR_MESSAGE</strong> : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Variables</td>
<td><strong>NB_LINE</strong> : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable Flow et retourne un entier. <strong>QUERY</strong> : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères. <strong>ERROR_MESSAGE</strong> : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace</td>
</tr>
</tbody>
</table>
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités des requêtes MDX pour des groupes de données multidimensionnels.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Ce composant requiert l’installation des fichiers .jar liés.</td>
</tr>
</tbody>
</table>

### Scénario : Tables en jointure croisée

Le présent Job permet d'extraire des groupes de données multidimensionnels à partir de tables de bases de données relationnelles stockées dans une base MySQL. Les données sont récupérées à l'aide d'une expression multidimensionnelle (requête MDX). Il est clair que vous devez connaître la structure de vos données, ou du moins disposer d'une description de la structure (catalogue) comme référence du groupe de données à récupérer dans ces différentes dimensions.

### Configurer le Job

**Procédure**

1. A partir de la Palette, cliquez-déposez un composant **tMondrianInput** (famille Business Intelligence) ainsi qu'un composant de sortie **tLogRow** (famille Logs & Errors).
2. Reliez le **tMondrianInput** au composant de sortie à l'aide d'un lien de type Row Main.

### Configurer la connexion à la base de données

**Procédure**

1. Double-cliquez sur le composant **tMondrianInput** pour afficher la vue Basic settings.
2. Dans le champ **DB Type**, sélectionnez la base de données relationnelle à utiliser avec Mondrian.

3. Sélectionnez l’entrée du **Repository** correspondante dans le champ **Property type**, dans le cas où vous stockez les détails de connexion à vos bases de données de manière centralisée. Dans cet exemple, les propriétés sont en mode **Built-in**.

4. Remplissez les détails de connexion à votre base de données : **Host**, **Port**, **Database**, **User Name** et **Password**.

5. Dans le champ **Schema**, sélectionnez le schéma correspondant à partir du **Repository** dans le cas où il est stocké de manière centralisée. Dans cet exemple, le schéma doit reste à définir (mode **Built-in**).

---

**Configurer la requête sur la base de données**

**Procédure**

1. La base de données relationnelle que l’on veut interroger contient cinq colonnes : **media**, **drink**, **unit_sales**, **store_cost** et **store_sales**.

2. La requête doit permettre de récupérer le contenu de **unit_sales**, **store_cost** et de **store_sales** pour les différentes valeur de **media** / **drink** en utilisant une requête MDX comme dans l’exemple ci-dessous : 

```sql
"select
{[Measures].[Unit Sales], [Measures].[Store Cost],
[Measures].[Store Sales]} on columns,
CrossJoin(
{[Promotion Media] [All Media] [Radio],
[Promotion Media] [All Media] [TV],
[Promotion Media] [All Media] [Sunday Paper],
[Promotion Media] [All Media] [Street Handout] },
[Product] [All Products] [Drink].children) on rows
from Sales
where ([Time] [1997])"
```
CrossJoin Example 1

The current slicer is 1997.

<table>
<thead>
<tr>
<th></th>
<th>Unit Sales</th>
<th>Store Cost</th>
<th>Store Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio</td>
<td>Alcohol</td>
<td>75</td>
<td>70.40</td>
</tr>
<tr>
<td></td>
<td>Beverages</td>
<td>97</td>
<td>75.70</td>
</tr>
<tr>
<td></td>
<td>Dairy</td>
<td>54</td>
<td>36.75</td>
</tr>
<tr>
<td>TV</td>
<td>Alcohol</td>
<td>76</td>
<td>70.99</td>
</tr>
<tr>
<td></td>
<td>Beverages</td>
<td>188</td>
<td>167.00</td>
</tr>
<tr>
<td></td>
<td>Dairy</td>
<td>60</td>
<td>45.19</td>
</tr>
<tr>
<td>Sunday Paper</td>
<td>Alcohol</td>
<td>148</td>
<td>128.97</td>
</tr>
<tr>
<td></td>
<td>Beverages</td>
<td>197</td>
<td>161.81</td>
</tr>
<tr>
<td></td>
<td>Dairy</td>
<td>85</td>
<td>54.75</td>
</tr>
<tr>
<td>Street Handout</td>
<td>Alcohol</td>
<td>158</td>
<td>121.14</td>
</tr>
<tr>
<td></td>
<td>Beverages</td>
<td>270</td>
<td>201.28</td>
</tr>
<tr>
<td></td>
<td>Dairy</td>
<td>84</td>
<td>50.26</td>
</tr>
</tbody>
</table>

3. A partir de l'onglet **Basic settings** du composant **tMondrianInput**, renseignez le chemin d'accès au catalogue de l'entrepôt de données dans le champ **Catalog**. Ce catalogue décrit la structure de l'entrepôt.

4. Entrez ensuite la requête MDX de la manière suivante :

   ```mdx
   "select
   ([Measures].[Unit Sales], [Measures].[Store Cost], [Measures].[Store Sales]) on columns,
   CrossJoin(
     { [Promotion Media].[All Media].[Radio],
       [Promotion Media].[All Media].[TV],
       [Promotion Media].[All Media].[Sunday Paper],
       [Promotion Media].[All Media].[Street Handout] },
     [Product].[All Products].[Drink].children) on rows
   from Sales
   where ([Time].[1997])"
   ``

5. Sélectionnez enfin le type d'encodage à partir de la liste du champ **Encoding**.

**Exécuter le Job**

**Procédure**

1. Sélectionnez le composant **tLogRow** puis cochez l'option **Print header** pour afficher le nom des colonnes sur la console.

2. Enregistrez le Job puis appuyez sur **F6** pour l’exécuter.
Résultats

La console affiche le résultat de `unit_sales`, `store_cost` et `store_sales` pour chaque type de `Drink` (`Beverages, Dairy, Alcoholic beverages`) croisé avec chaque type de `media` (`TV, Sunday Paper, Street handout`) comme illustré ci-dessus.
tMongoDBBulkLoad

Ce composant importe des fichiers de données dans différents formats (CSV, TSV ou JSON) dans la base de données MongoDB spécifiée, afin que les données puissent être traitées.

Propriétés du tMongoDBBulkLoad Standard

Ces propriétés sont utilisées pour configurer le tMongoDBBulkLoad s'exécutant dans le framework de Jobs Standard.

Le composant tMongoDBBulkLoad Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

| MongoDB directory | Saisissez dans ce champ le répertoire d’installation de MongoDB. |

| Use local DB path | Cochez cette case afin de fournir les informations de la base de données locale que vous souhaitez utiliser. Les versions 3.0 et supérieures de MongoDB ne supportent pas cette fonctionnalité.

- Local DB path : saisissez le chemin d’accès à la base de données locale spécifiée lors du démarrage du serveur MongoDB. |

| Use replica set address | Cochez la case pour afficher la table Replica address. |

- Replica set name : spécifiez le nom du Replica set. |
<table>
<thead>
<tr>
<th><strong>tMongoDBBulkLoad</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>Replica address</strong> : spécifiez les différents serveurs de la base de données MongoDB pour le failover. Si vous ne spécifiez rien dans les champs <strong>Replica host</strong> ou <strong>Replica port</strong> vides, leur valeur par défaut, respectivement <strong>localhost</strong> et <strong>27017</strong> sont utilisées.</td>
</tr>
</tbody>
</table>

| **Server** | Adresse IP et port d'écoute du serveur de la base de données. Notez que la valeur par défaut, **localhost**, est utilisée si le serveur n'est pas spécifié. Ce champ est disponible uniquement lorsque la case **Use replica set address** n'est pas cochée. |
| **Port** | Port d'écoute du serveur de la base de données. Notez que la valeur par défaut, **27017**, est utilisée si le port n'est pas spécifié. Ce champ est disponible uniquement lorsque la case **Use replica set address** n'est pas cochée. |
| **Database** | Saisissez le nom de la base de données dans laquelle importer les données. |
| **Collection** | Saisissez le nom de la collection dans laquelle importer les données. |
| **Use SSL connection** | Cochez cette case pour activer la connexion chiffrée SSL ou TLS. Utilisez le composant **tSetKeystore** dans le même Job afin de spécifier les informations de chiffrement. Pour plus d'informations concernant le **tSetKeystore**, consultez **tSetKeystore** à la page 3745. Notez que la connexion SSL est disponible uniquement à partir de la version 2.4 de MongoDB. |
| **Drop collection if exist** | Cochez cette case afin de supprimer la collection si elle existe déjà. |
| **Required authentication** | Cochez cette case pour activer l'authentification à la base de données. Parmi les mécanismes listés dans la liste déroulante **Authentication mechanism**, le mécanisme **NEGOTIATE** est recommandé si vous n'utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d'authentification le plus adapté à la version de MongoDB que vous utilisez. Pour plus d'informations sur les autres mécanismes de la liste, consultez **MongoDB Authentication** (en anglais) dans la documentation MongoDB. |
| **Set Authentication database** | Si le nom d'utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d'authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ **Authentication database** qui s'affiche. |
Pour plus d’informations sur la base de données d’authentification MongoDB, consultez User Authentication database (en anglais).

| Username et Password | Saisissez les informations d’authentification de l’utilisateur de la base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.  
Ces champs sont disponibles lorsque la case Required authentication est cochée.  
Si le système de sécurité sélectionné dans la liste Authentication mechanism est Kerberos, saisissez les informations dans les champs suivants User principal, Realm et KDC server et non dans les champs Username et Password. |
|----------------------|----------------------------------------------------------------------------------------------------------|
| Data file            | Saisissez le chemin d’accès complet au fichier duquel importer les données, ou cliquez sur le bouton [...] afin de parcourir votre système jusqu’au fichier de données.  
\[Avertissement\]  
Assurez-vous que le fichier de données est bien dans un format standard. Par exemple, les champs des fichiers CSV doivent être séparés par une virgule. |
| File type            | Sélectionnez le type de fichier dans la liste. Les formats CSV, TSV et JSON sont supportés. |
| The JSON file starts with an array | Cochez cette case pour permettre au tMongoDBBulkLoad de lire les fichiers JSON commençant par un tableau.  
Cette case est disponible lorsque vous avez sélectionné JSON dans la liste File type. |
| Action on data       | Sélectionnez l’action à effectuer sur les données.  
• Insert : insérer des données dans la base de données.  
\[Avertissement\]  
Lorsque vous insérez des données depuis des fichiers CSV ou TSV dans la base de données MongoDB, vous devez spécifier les champs, en cochant la case First line is header ou en les définissant dans le schéma.  
• Upsert : insérer les données si elles n’existent pas ou les mettre à jour si elles existent.  
\[Avertissement\]  
Lors d’une opération d’upsert de données dans la base de données MongoDB, vous devez spécifier une liste de champs, pour la partie requête de l’opération. |
| **Upsert fields** | Personnalisez les champs sur lesquels effectuer un upsert.  
Cette table est disponible lorsque vous sélectionnez **Upsert** dans la liste **Action on data**. |
|---|---|
| **First line is header** | Cochez cette case pour utiliser la première ligne d’un fichier CSV ou TSV en tant qu’en-tête.  
Cette case est disponible uniquement lorsque vous sélectionnez CSV ou TSV dans la liste **File type**. |
| **Ignore blanks** | Cochez cette case pour ignorer les champs vides dans les fichiers CSV ou TSV.  
Cette case est disponible uniquement lorsque vous sélectionnez CSV ou TSV dans la liste **File type**. |
| **Print log** | Cochez cette case pour afficher les logs. |

**Advanced settings**

| **Additional arguments** | Renseignez cette table afin d’utiliser des arguments supplémentaires selon vos besoins.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau des composants.</td>
</tr>
</tbody>
</table>

**Global Variables**

| **Global Variables** | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |
|---|---|
**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec un tMongoDBInput afin de vérifier si les données sont bien importées.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>L’outil client MongoDB doit être installé sur la machine où les Jobs contenant ce composant sont exécutés.</td>
</tr>
</tbody>
</table>

**Scénario : Importer des données dans la base de données MongoDB**

Ce scénario s'applique uniquement aux solutions Talend avec Big Data.

Le scénario suivant présente un Job qui importe des données d'un fichier CSV dans la collection spécifiée de MongoDB, puis lit les données de la collection MongoDB, afin de vérifier la réussite de l'import. Il continue à importer les données d'un fichier JSON ayant la même structure de données dans la même collection, puis affiche les données de la collection MongoDB afin de démontrer que les données du fichier JSON ont également été importées avec succès.
**Construire le Job**

**Procédure**

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : deux `tMongoDBBulkLoad`, deux `tMongoDBInput` et deux `tLogRow`.
2. Reliez le premier `tMongoDBBulkLoad` au premier `tMongoDBInput` à l'aide d'un lien `Trigger > OnSubjobOk`.
3. Reliez le premier `tMongoDBInput` au premier `tLogRow` à l'aide d'un lien `Row > Main`.
4. Répétez les deux étapes précédentes afin de connecter le second `tMongoDBBulkLoad` au second `tMongoDBInput` et le second `tMongoDBInput` au second `tLogRow`.
5. Reliez le premier `tMongoDBInput` au second `tMongoDBBulkLoad` à l'aide d'un lien `Trigger > OnSubjobOk`.
6. Renommez les deux composants `tLogRow` afin de mieux identifier les données affichées dans la console.

**Configurer les composants**

**Importer des données d’un fichier CSV**

**Procédure**

1. Double-cliquez sur le premier `tMongoDBBulkLoad` pour ouvrir sa vue **Basic settings**.

   ![Image du composant tMongoDBBulkLoad](image)

   2. Dans la liste **DB Version**, sélectionnez la version de MongoDB que vous utilisez.
   3. Dans les champs **Server** et **Port**, saisissez respectivement les informations relatives au serveur et au port, afin de vous connecter à MongoDB. Dans cet exemple, saisissez `localhost` et `27017`.
   4. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle importer les données, `bookstore` dans cet exemple.
Dans le champ **Collection**, saisissez la collection dans laquelle importer des données, *books* dans cet exemple.

5. Cochez la case **Drop collection if exist** afin de supprimer la collection spécifiée si elle existe déjà.


9. Cochez la case **First line is header** pour utiliser la première ligne du fichier CSV comme en-tête.

Cochez la case **Ignore blanks** pour ignorer les champs vides (s’il y en a) dans le fichier CSV.

Vérifier que le fichier CSV est bien importé

**Procédure**

1. Double-cliquez sur le composant *tMongoDBInput* pour ouvrir sa vue **Basic settings**.

   ![tMongoDBInput](image)

   2. Dans les champs **Server** et **Port**, saisissez les informations requises de connexion à MongoDB. Dans cet exemple, saisissez *localhost* et *27017*, respectivement.
3. Dans le champ **Database**, saisissez le nom de la base de données à lire, *bookstore* dans cet exemple.

4. Dans le champ **Collection**, saisissez la collection à partir de laquelle lire les données, *books* dans cet exemple.

5. Cliquez sur **Edit schema** pour définir la structure du schéma à lire de la collection MongoDB.

6. Dans la table **Mapping**, le champ **Column** est automatiquement renseigné avec le schéma défini. Ici, la colonne **Parent node path** ne doit pas nécessairement être remplie.

7. Double-cliquez sur le premier **tLogRow** pour ouvrir sa vue **Basic settings**.

8. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)**.
Importer des données d’un fichier JSON

Procédure

1. Double-cliquez sur le second tMongoDBBulkLoad pour ouvrir sa vue Basic settings.

2. Dans le champ MongoDB directory, saisissez le répertoire d’installation de MongoDB. Dans cet exemple, saisissez D:/MongoDB.

3. Dans les champs Server et Port, saisissez les informations requises pour la connexion à MongoDB. Dans cet exemple, saisissez respectivement localhost et 27017.

4. Dans le champ Database, saisissez le nom de la base de données dans laquelle importer des données, bookstore dans cet exemple. Dans le champ Collection, saisissez la collection cible dans laquelle importer des données, books dans cet exemple.


7. Sélectionnez Insert dans la liste Action on data.

Dans cet exemple, ajoutez l’argument "--jsonArray" pour importer les données sous forme de tableau JSON.

**Vérifier que le fichier JSON a bien été importé**

**Procédure**

1. Répétez les étapes 1 à 7 décrites dans la procédure Vérifier que le fichier CSV est bien importé à la page 2446 afin de configurer le second tMongoDBInput.

2. Répétez les étapes 8 et 9 décrites dans la procédure Vérifier que le fichier CSV est bien importé à la page 2446 afin de configurer le second tLogRow.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Exécutez le Job en appuyant sur F6 ou en cliquant sur le bouton Run de la vue Run.

Les données de la collection books dans la base de données MongoDB bookstore sont affichées dans le console et contiennent les données importées du fichier CSV books.csv et du fichier JSON books.json.
tMongoDBClose

Ce composant ferme une connexion à la base de données MongoDB.

**Propriétés du tMongoDBClose Standard**

Ces propriétés sont utilisées pour configurer le tMongoDBClose s’exécutant dans le framework de Jobs Standard.

Le composant tMongoDBClose Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Component list | Sélectionnez le composant tMongoDBConnection dans la liste, si plus d’une connexion est créée pour le Job. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants MongoDB, notamment le tMongoDBConnection. |
Scénario associé

Pour un scénario associé, consultez Scénario : Créer une collection et écrire des données dans celle-ci à la page 2498.
**tMongoDBConnection**

Ce composant crée une connexion à une base de données MongoDB et permet de réutiliser cette connexion dans d'autres composants.

Le tMongoDBConnection ouvre une connexion à une base de données afin d’effectuer une transaction.

**Propriétés du tMongoDBConnection Standard**

Ces propriétés sont utilisées pour configurer le tMongoDBConnection s’exécutant dans le framework de Jobs Standard.

Le composant tMongoDBConnection Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Version</td>
<td>Saisissez dans la liste la version de la base de données que vous utilisez.</td>
</tr>
<tr>
<td>Use replica set address</td>
<td>Cochez cette case pour afficher la table Replica address.</td>
</tr>
<tr>
<td></td>
<td>Dans la table Replica address, vous pouvez configurer différents serveurs de la base de données Mongo pour le failover.</td>
</tr>
<tr>
<td>Server et Port</td>
<td>Adresse IP et port d'écoute du serveur de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Ces champs sont disponibles lorsque les cases Use existing connection et Use replica set address ne sont pas cochées.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Use SSL connection</td>
<td>Cochez cette case pour activer la connexion chiffrée SSL ou TLS.</td>
</tr>
<tr>
<td></td>
<td>Utilisez le composant tSetKeystore dans le même Job afin de spécifier les informations de chifrement.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant le tSetKeystore, consultez tSetKeystore à la page 3745.</td>
</tr>
<tr>
<td></td>
<td>Notez que la connexion SSL est disponible uniquement à partir de la version 2.4 de MongoDB.</td>
</tr>
<tr>
<td>Required authentication</td>
<td>Cochez cette case pour activer l’authentification à la base de données.</td>
</tr>
<tr>
<td></td>
<td>Parmi les mécanismes listés dans la liste déroulante Authentication mechanism, le mécanisme NEGOTIATE est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez.</td>
</tr>
</tbody>
</table>
### Set Authentication database

Si le nom d'utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d'authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ **Authentication database** qui s'affiche.

Pour plus d'informations sur la base de données d'authentification MongoDB, consultez [User Authentication database](https://docs.mongodb.org/manual/core/authentication/) (en anglais).

### Username et Password

Informations d'authentification de l'utilisateur à la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Ces champs sont disponibles lorsque la case **Required authentication** est cochée.

Si le système de sécurité sélectionné dans la liste **Authentication mechanism** est **Kerberos**, saisissez les informations dans les champs suivants **User principal**, **Realm** et **KDC server** et non dans les champs **Username** et **Password**.

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

| No query timeout        | Cochez cette case pour empêcher les serveurs MongoDB de fermer les curseurs inactifs après 10 minutes d'inactivité de ces curseurs. Dans cette situation, un curseur inactif reste ouvert jusqu'à ce que les résultats de ce curseur soient épuisés ou que vous fermez manuellement le curseur à l'aide de la méthode `cursor.close()`.

Un curseur, dans MongoDB, est un pointeur vers l'ensemble de résultats de la requête. Par défaut, c'est-à-dire lorsque la case est décochée, un serveur MongoDB ferme automatiquement les curseurs inactifs après une période d'inactivité donnée, pour éviter les utilisations excessives de mémoire. Pour plus d'informations concernant les curseurs MongoDB, consultez [https://docs.mongodb.org/manual/core/cursors/](https://docs.mongodb.org/manual/core/cursors/) (en anglais). |

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. |
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants MongoDB, notamment le **tMongoClose**. |

**Scénario associé**

Pour un scénario associé, consultez **Scénario : Créer une collection et écrire des données dans celle-ci** à la page 2498.
tMongoDBGridFSDelete

Ce composant automatise la suppression de fichiers spécifiques dans le GridFS de MongoDB.

Le tMongoDBGridFSDelete se connecte à un système GridFS de MongoDB donné, exécute une requête définie par l'utilisateur pour sélectionner des fichiers spécifiques puis supprime ces fichiers du système.

Propriétés du tMongoDBGridFSDelete Standard

Ces propriétés sont utilisées pour configurer le tMongoDBGridFSDelete s'exécutant dans le framework de Jobs Standard.

Le composant tMongoDBGridFSDelete Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Property type       | Peut être Built-In ou Repository.  
|---------------------|----------------------------------|
|                     | Built-In : propriétés utilisées ponctuellement.  
|                     | Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.  

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.  
|-----------------------------| Notez que lorsqu'un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.  

| Use replica set address or multiple query routers | Cochez cette case pour afficher la table Server addresses.  
|--------------------------------------------------| Dans la table Server addresses, définissez les bases de données MongoDB partitionnées (sharded) ou les Replica sets de MongoDB auxquels vous souhaitez vous connecter.  

| Server et Port | Adresse IP et port d'écoute du serveur de la base de données.  
|----------------| Disponible lorsque la case Use replica set address est décochée.  
|                | Notez que si vous utilisez les mécanismes d'authentification pour vous connecter à la base de données MongoDB, vous devez saisir le nom de l'hôte du serveur de la base de données plutôt que son adresse IP.  

| Database | Nom de la base de données.  

| Use SSL connection | Cochez cette case pour activer la connexion chiffrée SSL ou TLS.  

Utilisez le composant `tSetKeystore` dans le même Job afin de spécifier les informations de chiffrement.

Pour plus d’informations concernant le `tSetKeystore`, consultez `tSetKeystore` à la page 3745.

Notez que la connexion SSL est disponible uniquement à partir de la version 2.4 de MongoDB.

| Set read preference | Cochez cette case et, dans la liste Read preference qui s’affiche, sélectionnez le membre auquel vous souhaitez adresser les opérations de lecture.
|                    | Si vous laissez décochée cette case, le Job utilise les préférences de lecture par défaut. Autrement dit, il utilise le membre primaire d’un Replica set.
|                    | Pour plus d’informations, consultez la documentation de MongoDB relative à la réplication et à ses préférences de lecture (Read preferences).

| Use authentication | Cochez cette case pour activer l’authentification à la base de données.
|                    | Parmi les mécanismes listés dans la liste déroulante Authentication mechanism, le mécanisme NEGOTIATE est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez.
|                    | Pour plus d’informations sur les autres mécanismes de la liste, consultez MongoDB Authentication (en anglais) dans la documentation MongoDB.

| Set Authentication database | Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ Authentication database qui s’affiche.
|                           | Pour plus d’informations sur la base de données d’authentification MongoDB, consultez User Authentication database (en anglais).

| Username et Password | Données d’authentification de l’utilisateur de la base de données.
|                     | Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.
|                     | Disponible lorsque la case Required authentication est cochée.
|                     | Si le système de sécurité sélectionné dans la liste Authentication mechanism est Kerberos, saisissez les informations dans les champs suivants User principal, Realm et KDC server et non dans les champs Username et Password.
<table>
<thead>
<tr>
<th><strong>Bucket</strong></th>
<th>Saisissez le nom du bucket à partir duquel supprimer les fichiers. Un bucket de GridFS est similaire à un dossier.</th>
</tr>
</thead>
</table>
| **Query type** | Sélectionnez le type de requête que le `tMongoDBGridFSDelete` doit exécuter pour sélectionner les fichiers à supprimer.  
- **Filename** : vous devez saisir le nom du fichier à supprimer. Ce nom est la valeur de l'attribut `filename` de ce fichier. Notez qu'un seul nom est autorisé.  
  Dans GridFS, un fichier se distingue seulement par son identifiant unique (son ID d'objet) ; son nom de fichier n'est pas forcément unique.  
- **Mongo Query** : saisissez la requête à utiliser pour sélectionner les fichiers à supprimer.  
  La requête d'exemple, `{}` entre guillemets doubles fournie par le `tMongoDBGridFSDelete`, sert à sélectionner tous les fichiers d'un bucket. Vous pouvez saisir `{'filename': {'$regex': 'REGEX_PATTERN'}}` pour appliquer une expression régulière afin de définir le nom des fichiers à utiliser. |

**Advanced settings**

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |
| **No query timeout** | Cochez cette case pour empêcher les serveurs MongoDB de fermer les curseurs inactifs après 10 minutes d'inactivité de ces curseurs. Dans cette situation, un curseur inactif reste ouvert jusqu'à ce que les résultats de ce curseur soient épuisés ou que vous fermez manuellement le curseur à l'aide de la méthode `cursor.close()`.  
  Un curseur, dans MongoDB, est un pointeur vers l'ensemble de résultats de la requête. Par défaut, c'est-à-dire lorsque la case est décochée, un serveur MongoDB ferme automatiquement les curseurs inactifs après une période d'inactivité donnée, pour éviter les utilisations excessives de mémoire. Pour plus d'informations concernant les curseurs MongoDB, consultez https://docs.mongodb.org/manual/core/cursors/ (en anglais). |

**Global Variables**

| **Global Variables** | **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable `After` et retourne un entier.  
**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case `Die on error` est décochée, si le composant a cette option. |
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour remplir un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

| Règle d’utilisation | Ce composant combine la connexion GridFS de MongoDB et l’itération de données. Il est ainsi souvent utilisé comme sous-job à un composant pour supprimer les fichiers sélectionnés d’un bucket donné dans le GridFS de MongoDB.


### Scénario associé

Pour un scénario utilisant le **tMongoDBGridFSDelete**, consultez **Gérer des fichiers en utilisant le GridFS de MongoDB** à la page 2475.
tMongoDBGridFSGet

Ce composant se connecte à un système GridFS de MongoDB pour copier des fichiers depuis celui-ci.

Le tMongoDBGridFSGet copie des fichiers depuis un système GridFS de MongoDB donné dans un répertoire local et renomme ces fichiers si nécessaire.

**Propriétés du tMongoDBGridFSGet Standard**

Ces propriétés sont utilisées pour configurer le tMongoDBGridFSGet s'exécutant dans le framework de Jobs Standard.

Le composant tMongoDBGridFSGet Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Property type | Peut être *Built-In* ou *Repository*.
  *Built-In*: propriétés utilisées ponctuellement.
  *Repository*: sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste *Component List* pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |

| Use replica set address or multiple query routers | Cochez cette case pour afficher la table *Server addresses*. |

| Database | Nom de la base de données. |

| Use SSL connection | Cochez cette case pour activer la connexion chiffrée SSL ou TLS. |
Utilisez le composant **tSetKeystore** dans le même Job afin de spécifier les informations de chiffrement.

Pour plus d’informations concernant le **tSetKeystore**, consultez **tSetKeystore** à la page 3745.

Notez que la connexion SSL est disponible uniquement à partir de la version 2.4 de MongoDB.

### Set read preference

Cochez cette case et, dans la liste **Read preference** qui s’affiche, sélectionnez le membre auquel vous souhaitez adresser les opérations de lecture.

Si vous laissez décochée cette case, le Job utilise les préférences de lecture par défaut. Autrement dit, il utilise le membre primaire d’un Replica set.

Pour plus d’informations, consultez la documentation de MongoDB relative à la réplication et à ses préférences de lecture (Read preferences).

### Required authentication

Cochez cette case pour activer l’authentification à la base de données.

Parmi les mécanismes listés dans la liste déroulante **Authentication mechanism**, le mécanisme **NEGOTIATE** est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez.

Pour plus d’informations sur les autres mécanismes de la liste, consultez MongoDB Authentication (en anglais) dans la documentation MongoDB.

### Set Authentication database

Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ **Authentication database** qui s’affiche.

Pour plus d’informations sur la base de données d’authentification MongoDB, consultez User Authentication database (en anglais).

### Username et Password

Données d’authentification de l’utilisateur de la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Disponible lorsque la case **Required authentication** est cochée.

Si le système de sécurité sélectionné dans la liste **Authentication mechanism** est **Kerberos**, saisissez les informations dans les champs suivants **User principal**, **Realm** et **KDC server** et non dans les champs **Username** et **Password**.
| **Bucket** | Saisissez le nom du bucket à partir duquel lire les fichiers. Un bucket de GridFS est similaire à un dossier. |
| **Query type** | Sélectionnez le type de requête que le tMongoDBGridFSDelete doit exécuter pour sélectionner les données à copier.  
- **Filename** : vous devez saisir le nom du fichier à copier. Ce nom est la valeur de l'attribut filename utilisé par ce fichier. Notez qu'un seul nom est autorisé.  
  Dans GridFS, un fichier se distingue seulement par son identifiant unique (son ID d'objet) ; son nom de fichier n'est pas forcément unique.  
- **Mongo Query** : saisissez la requête à utiliser pour sélectionner les fichiers à copier.  
  La requête par défaut, `{ }` entre guillemets doubles, fournis avec ce composant, permet de sélectionner tous les fichiers. Vous pouvez également appliquer une expression régulière en saisissant `{ 'filename': { '$regex': 'REGEX_PATTERN' }}` pour définir les noms des fichiers à utiliser.  
  Il est recommandé d'utiliser votre requête en cochant les cases **Overwrite local files** et **Use Document ID as output filename** pour éviter tout problème de conflit dû aux noms de fichiers. |
| **Local Folder** | Saisissez le chemin du dossier dans lequel écrire les données copiées depuis GridFS. |
| **Overwrite local files** | Cochez cette case pour écraser le fichier en utilisant le même nom de fichier dans le dossier local utilisé. Dans le cas contraire, une exception sera retournée si ce type de situation est rencontré. |
| **Use Document ID as output filename** | Cochez cette case pour renommer les fichiers entrants à la volée en utilisant les valeurs de leurs ID d'objet. Cela vous permet de garder des fichiers nommés de manière identique. |

**Advanced settings**

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |
| **No query timeout** | Cochez cette case pour empêcher les serveurs MongoDB de fermer les curseurs inactifs après 10 minutes d'inactivité de ces curseurs. Dans cette situation, un curseur inactif reste ouvert jusqu'à ce que les résultats de ce curseur soient épuisés ou que vous fermez manuellement le curseur à l'aide de la méthode cursor.close().  
  Un curseur, dans MongoDB, est un pointeur vers l'ensemble de résultats de la requête. Par défaut, c'est-à-
dire lorsque la case est décochée, un serveur MongoDB ferme automatiquement les curseurs inactifs après une période d’inactivité donnée, pour éviter les utilisations excessives de mémoire. Pour plus d’informations concernant les curseurs MongoDB, consultez https://docs.mongodb.org/manual/core/cursors/ (en anglais).

**Global Variables**

- **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable *After* et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur générée par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.
- Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.
- Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
- Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

- **Règle d’utilisation**
  - Ce composant combine la connexion GridFS de MongoDB et l’itération de données. Il est ainsi souvent utilisé comme sous-job à un composant pour copier des données depuis GridFS dans une répertoire local défini par l’utilisateur.

**Scénario associé**

Pour un scénario utilisant le tMongoDBGridFSGet, consultez Gérer des fichiers en utilisant le GridFS de MongoDB à la page 2475.
tMongoDBGridFSLList

Ce composant récupère une liste de fichiers en fonction d’une requête.
Le tMongoDBGridFSLList fait une boucle sur les fichiers d’un bucket dans le GridFS de MongoDB.

Propriétés du tMongoDBGridFSLList Standard

Ces propriétés sont utilisées pour configurer le tMongoDBGridFSLList s’exécutant dans le framework de Jobs Standard.
Le composant tMongoDBGridFSLList Standard appartient aux familles Big Data et Databases.
Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Property type | Peut être Built-In ou Repository. 
| Built-In : propriétés utilisées ponctuellement. 
| Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. 
| Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
| Use replica set address or multiple query routers | Cochez cette case pour afficher la table Server addresses. 
| Dans la table Server addresses, définissez les bases de données MongoDB partitionnées (sharded) ou les Replica sets de MongoDB auxquels vous souhaitez vous connecter. |
| Server et Port | Adresse IP et port d’écoute du serveur de la base de données. 
| Disponible lorsque la case Use replica set address est décochée. 
| Notez que si vous utilisez les mécanismes d’authentification pour vous connecter à la base de données MongoDB, vous devez saisir le nom de l’hôte du serveur de la base de données plutôt que son adresse IP. |
| Database | Nom de la base de données. |
| Use SSL connection | Cochez cette case pour activer la connexion chiffrée SSL ou TLS. 
| Utilisez le composant tSetKeystore dans le même Job afin de spécifier les informations de chiffrement. |
Pour plus d’informations concernant le `tSetKeystore`, consultez `tSetKeystore` à la page 3745.
Notez que la connexion SSL est disponible uniquement à partir de la version 2.4 de MongoDB.

<table>
<thead>
<tr>
<th>Set read preference</th>
</tr>
</thead>
</table>
| Cochez cette case et, dans la liste Read preference qui s’affiche, sélectionnez le membre auquel vous souhaitez adresser les opérations de lecture.  
Si vous laissez décochée cette case, le Job utilise les préférences de lecture par défaut. Autrement dit, il utilise le membre primaire d’un Replica set.  
Pour plus d’informations, consultez la documentation de MongoDB relative à la réplication et à ses préférences de lecture (Read preferences). |

<table>
<thead>
<tr>
<th>Use authentication</th>
</tr>
</thead>
</table>
| Cochez cette case pour activer l’authentification à la base de données.  
Parmi les mécanismes listés dans la liste déroulante Authentication mechanism, le mécanisme NEGOTIATE est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez.  
Pour plus d’informations sur les autres mécanismes de la liste, consultez MongoDB Authentication (en anglais) dans la documentation MongoDB. |

<table>
<thead>
<tr>
<th>Set Authentication database</th>
</tr>
</thead>
</table>
| Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ Authentication database qui s’affiche.  
Pour plus d’informations sur la base de données d’authentification MongoDB, consultez User Authentication database (en anglais). |

<table>
<thead>
<tr>
<th>Username et Password</th>
</tr>
</thead>
</table>
| Données d’authentification de l’utilisateur de la base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.  
Disponible lorsque la case Required authentication est cochée.  
Si le système de sécurité sélectionné dans la liste Authentication mechanism est Kerberos, saisissez les informations dans les champs suivants User principal, Realm et KDC server et non dans les champs Username et Password. |
### Bucket
Saisissez le nom du bucket dans lequel les fichiers à lister sont stockés. Un bucket de GridFS est similaire à un dossier.

### Query
Saisissez la requête à utiliser pour sélectionner les fichiers à lister. Vous devez définir la manière dont ces fichiers sont triés dans la liste en sélectionnant :

- **Sort by** : les fichiers peuvent être triés selon leurs attributs *filename*, *uploadDate* ou *length* (*l’option File Size*).
- **Sort order** : les fichiers peuvent être triés dans l’ordre *Ascending* (ascendant) ou *Descending* (descendant).

### Advanced settings

<table>
<thead>
<tr>
<th><strong>tStatCatcher Statistics</strong></th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

| **No query timeout** | Cochez cette case pour empêcher les serveurs MongoDB de fermer les curseurs inactifs après 10 minutes d’inactivité de ces curseurs. Dans cette situation, un curseur inactif reste ouvert jusqu’à ce que les résultats de ce curseur soient épuisés ou que vous fermez manuellement le curseur à l’aide de la méthode `cursor.close()`.

Un curseur, dans MongoDB, est un pointeur vers l’ensemble de résultats de la requête. Par défaut, c’est-à-dire lorsque la case est décochée, un serveur MongoDB ferme automatiquement les curseurs inactifs après une période d’inactivité donnée, pour éviter les utilisations excessives de mémoire. Pour plus d’informations concernant les curseurs MongoDB, consultez [https://docs.mongodb.org/manual/core/cursors/](https://docs.mongodb.org/manual/core/cursors/) (en anglais).

<table>
<thead>
<tr>
<th><strong>Global Variables</strong></th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Global Variables</strong></th>
<th><strong>NB_FILE</strong> : Indique le nombre de fichiers traités. Cette variable est une variable <em>After</em> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td><strong>CURRENT_FILEID</strong> : valeur de l’attribut <em>ObjectId</em> (l’ID du fichier) du fichier courant. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>CURRENT_FILENAME</strong> : valeur de l’attribut <em>filename</em> du fichier courant. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>CURRENT_FILEUPLOADDATE</strong> : valeur de l’attribut <em>uploadDate</em> du fichier courant. Cette variable est une variable <em>Flow</em> et retourne une date.</td>
</tr>
</tbody>
</table>
**CURRENT_FILELENGTH** : valeur de l’attribut `length` (taille) du fichier courant. Cette variable est une variable Flow et retourne une longueur.

**CURRENT_FILEMD5** : valeur de l’attribut `md5` du fichier courant. Cette variable est une variable Flow et retourne une chaîne de caractères.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

---

**Utilisation**

| Règle d’utilisation | Ce composant combine la connexion GridFS de MongoDB et l’itération de données. Il est ainsi souvent utilisé comme sous-job à un composant pour lister les fichiers stockés dans un bucket donné dans le GridFS de MongoDB.

Différentes listes sont retournées, chacune représente un attribut de ces fichiers et est stockée dans une variable globale spécifique comme expliqué ci-dessus concernant les variables globales fournies par le `tMongoDBGridFSList`.

Il est souvent relié au Job à l’aide d’un lien `OnSubjobOk` ou `OnComponentOk`, en fonction du contexte. |

---

**Scénario associé**

Pour un scénario utilisant le `tMongoDBGridFSList`, consultez Gérer des fichiers en utilisant le GridFS de MongoDB à la page 2475.
tMongoDBGridFSProperties

Ce composant récupère des informations concernant les propriétés de fichiers sélectionnés en fonction d’une requête.

Le tMongoDBGridFSProperties effectue une boucle sur les fichiers d’un bucket dans le GridFS de MongoDB pour extraire les attributs de ces fichiers.

Propriétés du tMongoDBGridFSProperties Standard

Ces propriétés sont utilisées pour configurer le tMongoDBGridFSProperties s’exécutant dans le framework de Jobs Standard.

Le composant tMongoDBGridFSProperties Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Property type       | Peut être Built-In ou Repository.  
|                     | Built-In : propriétés utilisées ponctuellement. Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
|                     | Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
| Use replica set address or multiple query routers | Cochez cette case pour afficher la table Server addresses.  
|                     | Dans la table Server addresses, définissez les bases de données MongoDB partitionnées (sharded) ou les Replica sets de MongoDB auxquels vous souhaitez vous connecter. |
| Server et Port      | Adresse IP et port d’écoute du serveur de la base de données.  
|                     | Disponible lorsque la case Use replica set address est décochée.  
|                     | Notez que si vous utilisez les mécanismes d’authentication pour vous connecter à la base de données MongoDB, vous devez saisir le nom de l’hôte du serveur de la base de données plutôt que son adresse IP. |
| Database            | Nom de la base de données. |
| Use SSL connection  | Cochez cette case pour activer la connexion chiffrée SSL ou TLS. |
Utilisez le composant **tSetKeystore** dans le même Job afin de spécifier les informations de chiffrement.

Pour plus d’informations concernant le **tSetKeystore**, consultez [tSetKeystore](#) à la page 3745.

Notez que la connexion SSL est disponible uniquement à partir de la version 2.4 de MongoDB.

**Set read preference**

Cochez cette case et, dans la liste **Read preference** qui s’affiche, sélectionnez le membre auquel vous souhaitez adresser les opérations de lecture.

Si vous laissez décochée cette case, le Job utilise les préférences de lecture par défaut. Autrement dit, il utilise le membre primaire d’un Replica set.

Pour plus d’informations, consultez la documentation de MongoDB relative à la réplication et à ses préférences de lecture (Read preferences).

**Use authentication**

Cochez cette case pour activer l’authentification à la base de données.

Parmi les mécanismes listés dans la liste déroulante **Authentication mechanism**, le mécanisme **NEGOTIATE** est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez.

Pour plus d’informations sur les autres mécanismes de la liste, consultez [MongoDB Authentication](#) dans la documentation MongoDB.

**Set Authentication database**

Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ **Authentication database** qui s’affiche.

Pour plus d’informations sur la base de données d’authentification MongoDB, consultez [User Authentication database](#) (en anglais).

**Username et Password**

Données d’authentification de l’utilisateur de la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Disponible lorsque la case **Required authentication** est cochée.

Si le système de sécurité sélectionné dans la liste **Authentication mechanism** est **Kerberos**, saisissez les informations dans les champs suivants **User principal**, **Realm** et **KDC server** et non dans les champs **Username** et **Password**.
### Bucket
Saisissez le nom du bucket dans lequel les fichiers à utiliser sont stockés. Un bucket de GridFS est similaire à un dossier.

### Schema et Edit schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Les colonnes du schéma prédéfini du `tMongoDBGridFS Properties` représentent les attributs par défaut d’un fichier GridFS et le nom de ces colonnes sont déjà identiques aux noms des attributs.

- Si vous souhaitez exclure certains attributs du flux de données à traiter dans votre Job, vous pouvez supprimer les colonnes correspondant à ces attributs à partir de ce schéma.
- Ne modifiez pas le nom des colonnes à utiliser car chaque nom de colonne doit être identique au nom de l’attribut du fichier qu’il représente.
- Si les fichiers à traiter contiennent des attributs personnalisés que vous souhaitez extraire, vous devez ajouter leurs colonnes respectives et vous assurer que ces colonnes et que ces attributs personnalisés utilisent des noms identiques.

### Query type
Sélectionnez le type de requête que le `tMongoDBGridFS Properties` doit exécuter pour sélectionner les fichiers à partir desquels extraire les attributs.

- **Filename** : saisissez le nom du fichier à utiliser. Ce nom est la valeur de l’attribut `filename` de ce fichier. Notez qu’un seul nom est autorisé.
  
  Dans GridFS, un fichier se distingue seulement par son identifiant unique (son ID d’objet) ; son nom de fichier n’est pas forcément unique.

- **Mongo Query** : saisissez la requête pour sélectionner les fichiers à utiliser.

  La requête d’exemple, `{}` entre guillemets doubles fournie par le `tMongoDBGridFS Properties`, sert à sélectionner tous les fichiers d’un bucket. Vous pouvez saisir `{'filename': {'$regex': 'REGEX_PATTERN'}}` pour appliquer une expression régulière afin de définir le nom des fichiers à utiliser.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>tStatCatcher Statistics</strong></th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No query timeout</strong></td>
<td>Cochez cette case pour empêcher les serveurs MongoDB de fermer les curseurs inactifs après 10 minutes d’inactivité de ces curseurs. Dans cette situation, un curseur inactif reste ouvert jusqu’à ce que les résultats</td>
</tr>
</tbody>
</table>
de ce curseur soient épuisés ou que vous fermiez manuellement le curseur à l’aide de la méthode `cursor.close()`.

Un curseur, dans MongoDB, est un pointeur vers l’ensemble de résultats de la requête. Par défaut, c’est-à-dire lorsque la case est décochée, un serveur MongoDB ferme automatiquement les curseurs inactifs après une période d’inactivité donnée, pour éviter les utilisations excessives de mémoire. Pour plus d’informations concernant les curseurs MongoDB, consultez [https://docs.mongodb.org/manual/core/cursors/](https://docs.mongodb.org/manual/core/cursors/) (en anglais).

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_FILE</strong></td>
<td>Indique le nombre de fichiers traités. Cette variable est une variable <code>After</code> et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <code>After</code> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <code>Die on error</code> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable `Flow` fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le [Guide utilisateur du Studio Talend](https://docs.talend.com/).

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant combine la connexion GridFS de MongoDB et l’extraction de données. Il nécessite d’être relié à l’aide d’un lien <code>Main</code> au composant qui le suit afin de lui envoyer des données.</td>
<td></td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario utilisant le `tMongoDBGridFSProperties`, consultez [Gérer des fichiers en utilisant le GridFS de MongoDB à la page 2475](#).
tMongoDBGridFSPut

Ce composant se connecte à un système GridFS de MongoDB pour y charger des fichiers.

Le tMongoDBGridFSPut copie des fichiers à partir d'un répertoire local, les colle dans un système GridFS de MongoDB donné et les renomme si nécessaire.

Propriétés du tMongoDBGridFSPut Standard

Ces propriétés sont utilisées pour configurer le tMongoDBGridFSPut s'exécutant dans le framework de Jobs Standard.

Le composant tMongoDBGridFSPut Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Built-In | Propriétés utilisées ponctuellement.  
Repository | Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.  
Notez que lorsqu'un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.

Use replica set address or multiple query routers

Cochez cette case pour afficher la table Server addresses.  
Dans la table Server addresses, définissez les bases de données MongoDB partitionnées (sharded) ou les Replica sets de MongoDB auxquels vous souhaitez vous connecter.

Server et Port

Adresse IP et port d'écoute du serveur de la base de données.  
Disponible lorsque la case Use replica set address est décochée.  
Notez que si vous utilisez les mécanismes d'authentification pour vous connecter à la base de données MongoDB, vous devez saisir le nom de l'hôte du serveur de la base de données plutôt que son adresse IP.

Database

Nom de la base de données.

Use SSL connection

Cochez cette case pour activer la connexion chiffrée SSL ou TLS.
Utilisez le composant `tSetKeystore` dans le même Job afin de spécifier les informations de chiffrément.

Pour plus d’informations concernant le `tSetKeystore`, consultez `tSetKeystore` à la page 3745.

Notez que la connexion SSL est disponible uniquement à partir de la version 2.4 de MongoDB.

<table>
<thead>
<tr>
<th>Set read preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case et, dans la liste <strong>Read preference</strong> qui s’affiche, sélectionnez le membre auquel vous souhaitez adresser les opérations de lecture.</td>
</tr>
<tr>
<td>Si vous laissez décochée cette case, le Job utilise les préférences de lecture par défaut. Autrement dit, il utilise le membre primaire d’un Replica set.</td>
</tr>
<tr>
<td>Pour plus d’informations, consultez la documentation de MongoDB relative à la réplication et à ses préférences de lecture (Read preferences).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Required authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour activer l’authentification à la base de données.</td>
</tr>
<tr>
<td>Parmi les mécanismes listés dans la liste déroulante <strong>Authentication mechanism</strong>, le mécanisme <strong>NEGOTIATE</strong> est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez.</td>
</tr>
<tr>
<td>Pour plus d’informations sur les autres mécanismes de la liste, consultez <strong>MongoDB Authentication</strong> (en anglais) dans la documentation MongoDB.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set Authentication database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ <strong>Authentication database</strong> qui s’affiche.</td>
</tr>
<tr>
<td>Pour plus d’informations sur la base de données d’authentification MongoDB, consultez <strong>User Authentication database</strong> (en anglais).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Username et Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Données d’authentification de l’utilisateur de la base de données.</td>
</tr>
<tr>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Disponible lorsque la case <strong>Required authentication</strong> est cochée.</td>
</tr>
<tr>
<td>Si le système de sécurité sélectionné dans la liste <strong>Authentication mechanism</strong> est <strong>Kerberos</strong>, saisissez les informations dans les champs suivants <strong>User principal</strong>, <strong>Realm</strong> et <strong>KDC server</strong> et non dans les champs <strong>Username</strong> et <strong>Password</strong>.</td>
</tr>
</tbody>
</table>
**Bucket**
Saisissez le nom du bucket à partir duquel supprimer les fichiers. Un bucket de GridFS est similaire à un dossier.

**Local Folder**
Parcourez votre système ou saisissez le nom du dossier dans lequel les fichiers à copier et à écrire dans GridFS sont stockés.

**Use Perl5 Regex Expression as Filemask**
Cochez cette case si vous souhaitez utiliser les expressions régulières Perl5 comme filtres de fichiers dans le champ **Files**. Cela est utile lorsque le nom du fichier à utiliser contient des caractères spéciaux comme des parenthèses.

Pour plus d’informations concernant la syntaxe des expressions régulières Perl5, consultez Perl5 Regular Expression Syntax (en anglais).

**Files**
Dans la zone **Files**, les champs à renseigner sont :
- **File mask** : saisissez le nom du fichier à sélectionner depuis le répertoire local. L’utilisation d’une expression régulière est possible.
- **New name** : donnez un nouveau nom au fichier chargé.

### Advanced settings

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

**No query timeout**
Cochez cette case pour empêcher les serveurs MongoDB de fermer les curseurs inactifs après 10 minutes d’inactivité de ces curseurs. Dans cette situation, un curseur inactif reste ouvert jusqu’à ce que les résultats de ce curseur soient épuisés ou que vous fermez manuellement le curseur à l’aide de la méthode `cursor.close()`.

Un curseur, dans MongoDB, est un pointeur vers l’ensemble de résultats de la requête. Par défaut, c’est-à-dire lorsque la case est décochée, un serveur MongoDB ferme automatiquement les curseurs inactifs après une période d’inactivité donnée, pour éviter les utilisations excessives de mémoire. Pour plus d’informations concernant les curseurs MongoDB, consultez https://docs.mongodb.org/manual/core/cursors/ (en anglais).

### Global Variables

**Global Variables**

- **NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable `After` et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case `Die on error` est décochée, si le composant a cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | Ce composant combine la connexion GridFS de MongoDB et l’extraction de données. Il est ainsi souvent utilisé comme sous-job à un composant pour copier des données depuis un répertoire local défini par l’utilisateur dans GridFS. Il est souvent relié au Job à l’aide d’un lien **OnSubjobOk** ou **OnComponentOk**, en fonction du contexte. |

## Gérer des fichiers en utilisant le GridFS de MongoDB

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Dans ce scénario, les composants GridFS de MongoDB sont utilisés pour créer un Job afin de gérer des fichiers vidéo dans le GridFS de MongoDB.

Pour plus d’informations sur le système GridFS de MongoDB, consultez *When to use GridFS* (en anglais).

Dans un but de démonstration, un seul fichier vidéo, nommé *custom_hadoop.mp4*, est utilisé ; vous pouvez utiliser un de vos propres fichiers vidéo pour reproduire ce scénario.
Relier les composants

Procédure

1.Dans la perspective **Integration** du Studio, créez un Job vide, nommé **FS_video** par exemple, depuis le nœud **Job Designs** dans la vue **Repository**.
   Pour plus d’informations sur la création d’un Job, consultez le Guide utilisateur du Studio Talend.

2. Dans l’espace de modélisation graphique, saisissez le nom du composant à utiliser et sélectionnez ce composant dans la liste qui s’affiche. Dans ce scénario, les composants sont le **tMongoDBConnection**, le **tMongoDBGridFSPut**, le **tMongoDBGridFSList**, le **tMongoDBGridFSProperties**, le **tFilterColumns**, le **tLogRow**, le **tMongoDBGridFSGet** et le **tMongoDBGridFSDelete**.

3. Reliez le **tMongoDBConnection** au **tMongoDBGridFSPut** à l’aide d’un lien **Trigger > OnSubjobOk**.

4. Répétez cette opération pour relier le **tMongoDBGridFSPut** au **tMongoDBGridFSList**, le **tMongoDBGridFSList** au **tMongoDBGridFSGet** puis le **tMongoDBGridFSGet** au **tMongoDBGridFSDelete**.

5. Reliez le **tMongoDBGridFSList** au **tMongoDBGridFSProperties** à l’aide d’un lien **Row > Iterate**. Ce lien permet au **tMongoDBGridFSList** d’envoyer des données au **tMongoDBGridFSProperties** de manière itérative.

6. Reliez le **tMongoDBGridFSProperties** au **tFilterColumns** à l’aide d’un lien **Row > Main**.

7. Faites de même pour relier le **tFilterColumns** au **tLogRow**.
Se connecter à MongoDB

Procédure

1. Double-cliquez sur le **tMongoDBConnection** pour ouvrir sa vue **Component**.

![tMongoDBConnection](image)

2. Dans la liste **DB version**, sélectionnez la version de MongoDB que vous utilisez.

3. Dans les champs **Server** et **Port**, saisissez les informations d’authentification requises pour vous connecter à MongoDB.

   Si vous utilisez le nom d’hôte du serveur MongoDB, assurez-vous d’avoir ajouté le mapping entre ce nom d’hôte et son adresse IP dans le fichier `hosts` du système d’exploitation dans lequel le Job actuel est exécuté.

4. Dans le champ **Database**, saisissez le nom de la base de données hébergeant GridFS. Cette base de données est créée à la volée si elle n’existe pas.

Copier des données dans le GridFS de MongoDB

Procédure

1. Double-cliquez sur le **tMongoDBGridFSPut** pour ouvrir sa vue **Component**.
2. Cochez la case **Use existing connection** et dans la liste **Connection**, sélectionnez le composant dans lequel la connexion à MongoDB à utiliser est définie.

3. Dans le champ **Bucket**, saisissez le bucket à utiliser pour stocker les fichiers dans GridFS. Dans cet exemple, saisissez `talend_channel/61`.

4. Dans le champ **Local folder**, saisissez le chemin ou parcourez votre système jusqu'au dossier dans lequel les fichiers à charger dans GridFS sont stockés. Comme expliqué précédemment, il s'agit d'un fichier vidéo nommé `custom_hadoop.mp4`.

5. Dans la table **Files**, ajoutez une ligne en cliquant sur le bouton `[+]` et dans la colonne **Filemask**, saisissez `*.mp4` entre guillemets doubles. Cela permet au **tMongoDBGridFSPut** de copier tous les fichiers avec l'extension `.mp4` depuis le dossier local que vous avez spécifié dans le bucket à utiliser dans GridFS.

6. Laissez la colonne **New name** vide, c'est-à-dire, laissez les guillemets doubles tels quels dans cette colonne, afin que cette vidéo garde le même nom après avoir été copiée dans GridFS.

**Lister des fichiers stockés dans le GridFS de MongoDB**

**Effectuer une boucle sur les fichiers**

**Procédure**

1. Double-cliquez sur le **tMongoDBGridFSList** pour ouvrir sa vue **Component**.
2. Cochez la case **Use existing connection** et dans la liste **Connection**, sélectionnez le composant dans lequel la connexion à MongoDB à utiliser est définie.

3. Dans le champ **Bucket**, saisissez le bucket dans lequel les fichiers à lister sont stockés. Dans cet exemple, saisissez *talend_channel/61*.

4. Dans le champ **Query**, saisissez la requête pour sélectionner les fichiers sur lesquels vous souhaitez que le tMongoDBGridFSList effectue une boucle afin de générer différentes listes de fichiers. Dans cet exemple, laissez celle par défaut pour effectuer une boucle sur tous les fichiers stockés dans le bucket *talend_channel/61*.

Comme expliqué précédemment, seul un fichier, *custom_hadoop.mp4*, est attendu.

### Extraire des métadonnées de fichier

**Procédure**

1. Double-cliquez sur le tMongoDBGridFSProperties pour ouvrir sa vue **Component**.

2. Cochez la case **Use existing connection** et dans la liste **Connection**, sélectionnez le composant dans lequel la connexion à MongoDB à utiliser est définie.
3. Dans le champ **Bucket**, saisissez le bucket dans lequel les fichiers à utiliser sont stockés. Dans cet exemple, saisissez `talend_channel/61`.

4. Dans la liste **Query type**, sélectionnez l’approche que vous souhaitez utiliser pour sélectionner les fichiers à partir desquels extraire les métadonnées. Dans cet exemple, sélectionnez **Filename** pour utiliser l’attribut `filename` de chaque fichier GridFS lors de chaque requête.

5. Dans le champ **Filename**, appuyez sur Ctrl + Espace pour afficher la liste des variables et choisir la variable à utiliser. Dans cet exemple, sélectionnez `tMongoDBGridFSList.CURRENT_FILENAME` dans la liste. L’expression pour utiliser la variable `CURRENT_FILENAME` est ensuite automatiquement ajoutée.

   Elle permet au `tMongoDBGridFSProperties` de lire chaque nom de fichier retourné par le `tMongoDBGridFSList`.

**Filtrer des attributs**

**Procédure**

1. Double-cliquez sur le `tFilterColumns` pour ouvrir sa vue **Component**.

   ![tFilterColumns](image)

2. Cliquez sur le bouton `[...]` à côté de **Edit schema** pour ouvrir l’éditeur de schéma.

3. Du côté gauche (côté de l’entrée), sélectionnez la colonne à utiliser et cliquez sur le bouton ![flèche](image) pour déplacer cette colonne du côté droit (côté de la sortie). Dans cet exemple, déplacez chaque colonne du côté droit sauf la colonne **contentType**.

   Chaque colonne représente un attribut de fichier et le schéma prédéfini du `tMongoDBGridFSProperties` contient automatiquement ces colonnes.
4. Cliquez sur OK pour valider ces modifications et accepter la propagation proposée par la boîte de dialogue qui s’ouvre.

**Télécharger des fichiers depuis le GridFS de MongoDB**

**Procédure**

1. Double-cliquez sur le tMongoDBGridFSGet pour ouvrir sa vue Component.

![tMongoDBGridFSGet_1](image)

2. Cochez la case **Use existing connection** et dans la liste Connection, sélectionnez le composant dans lequel la connexion à MongoDB à utiliser est définie.

3. Dans le champ **Bucket**, saisissez le bucket dans lequel sont stockés les fichiers à récupérer. Dans cet exemple, saisissez `talend_channel/61`.

4. Dans le champ **Local folder**, saisissez le chemin du dossier local dans lequel vous souhaitez stocker les fichiers téléchargés. Dans ce scénario, saisissez `C:/tmp/output`.

5. Cochez la case **Use Document ID as output filename** pour renommer chaque fichier téléchargé en utilisant la valeur de son attribut `ObjectID`.

Puisqu’un fichier dans GridFS se distingue par son ID plutôt que par son nom, il est possible que plusieurs fichiers possèdent le même nom. Pour cette raison, lorsque vous téléchargez ce type de fichiers dans le même répertoire sans les renommer différemment, une exception est retournée pour vous avertir que le fichier en cours de téléchargement existe déjà. Afin d’éviter cette erreur, vous pouvez soit cocher la case **Overwrite local files** pour remplacer le fichier existant par le dernier fichier téléchargé, soit renommer ces fichiers à la volée en utilisant leur ID. Dans cet exemple, c’est la stratégie de renommage des fichiers qui est adoptée.

**Supprimer des fichiers du GridFS de MongoDB**

**Procédure**

1. Double-cliquez sur le tMongoDBGridFSDelete pour ouvrir sa vue Component.
2. Cochez la case **Use existing connection** et dans la liste **Connection**, sélectionnez le composant dans lequel la connexion à MongoDB à utiliser est définie.


4. Dans la liste **Query type**, sélectionnez l’approche que vous souhaitez utiliser pour sélectionner les fichiers à supprimer. Dans cet exemple, sélectionnez **Filename** pour utiliser l’attribut `filename` de chaque fichier GridFS lors de la requête.

5. Dans le champ **Filename**, saisissez le nom du fichier à supprimer.

**Exécuter le Job**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez ensuite exécuter le Job.

Le composant **tLogRow** est utilisé pour présenter les résultats d’exécution du Job.

**Procédure**

1. Si vous souhaitez configurer le mode d’affichage dans sa vue **Component**, double-cliquez sur le composant **tLogRow** pour ouvrir la vue **Component** et dans la zone **Mode**, sélectionnez l’option Table (print values in cells of a table).

2. Appuyez sur **F6** pour exécuter le Job.

**Résultats**

Une fois l’exécution terminée, la vue **Run** s’ouvre automatiquement et affiche les métadonnées de la vidéo `custom_hadoop.mp4` dans GridFS.
Le fichier téléchargé se trouve dans le répertoire C:/tmp/output, son ID utilisée comme nom de fichier.
**tMongoDBInput**

Ce composant récupère des documents d’une collection, dans la base de données MongoDB et de les transférer au composant suivant, pour affichage ou stockage.

Le tMongoDBInput récupère des documents d’une collection de la base de données MongoDB en fournissant un document de requête contenant les champs auxquels doivent correspondre les documents souhaités.

**Propriétés du tMongoDBInput Standard**

Ces propriétés sont utilisées pour configurer le tMongoDBInput s’exécutant dans le framework de Jobs Standard.

Le composant tMongoDBInput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>DB Version</strong></td>
<td>Sélectionnez dans la liste la version de la base de données que vous utilisez. Cette option est disponible lorsque la case Use existing connection est décochée.</td>
</tr>
<tr>
<td><strong>Use replica set address</strong></td>
<td>Cochez cette case pour afficher la table Replica address. Dans la table Replica address, vous pouvez configurer différents serveurs de la base de données MongoDB pour le failover. Cette option est disponible lorsque la case Use existing connection n’est pas cochée.</td>
</tr>
<tr>
<td><strong>Server et Port</strong></td>
<td>Adresse IP et port d’écoute du serveur de la base de données. Ces champs sont disponibles lorsque les cases Use existing connection et Use replica set address ne sont pas cochées.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Use SSL connection</strong></td>
<td>Cochez cette case pour activer la connexion chiffrée SSL ou TLS. Utilisez le composant tSetKeystore dans le même Job afin de spécifier les informations de chiffrement.</td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant le `tSetKeystore`, consultez `tSetKeystore` à la page 3745. Notez que la connexion SSL est disponible uniquement à partir de la version 2.4 de MongoDB.

### Set read preference

Cochez cette case et, dans la liste `Read preference` qui s’affiche, sélectionnez le membre auquel vous souhaitez adresser les opérations de lecture.

Si vous laissez décochée cette case, le Job utilise les préférences de lecture par défaut. Autrement dit, il utilise le membre primaire d’un Réplica set.

Pour plus d’informations, consultez la documentation de MongoDB relative à la réplication et à ses préférences de lecture (`Read preferences`).

### Required authentication

Cochez cette case pour activer l’authentification à la base de données.

Parmi les mécanismes listés dans la liste déroulante `Authentication mechanism`, le mécanisme `NEGOTIATE` est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez.

Pour plus d’informations sur les autres mécanismes de la liste, consultez `MongoDB Authentication` (en anglais) dans la documentation MongoDB.

### Set Authentication database

Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ `Authentication database` qui s’affiche.

Pour plus d’informations sur la base de données d’authentification MongoDB, consultez `User Authentication database` (en anglais).

### Username et Password

Informations d’authentification de l’utilisateur à la base de données.

Pour saisir le mot de passe, cliquez sur le bouton `...` à côté du champ `Password`, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur `OK` afin de sauvegarder les paramètres.

Ces champs sont disponibles lorsque la case `Required authentication` est cochée.

Si le système de sécurité sélectionné dans la liste `Authentication mechanism` est Kerberos, saisissez les informations dans les champs suivants `User principal`, `Realm` et `KDC server` et non dans les champs `Username` et `Password`.

### Collection

Nom de la collection dans la base de données MongoDB.
### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

Si une colonne dans une base de données est un document JSON et que vous devez lire le document entier, saisissez un astérisque `*` dans la colonne **DB column**, sans guillemet.

### Query

Spécifiez la condition de la requête. Ce champ est disponible uniquement lorsque vous avez sélectionné **Find query** dans la liste **Query type**.

Par exemple, saisissez `"{id:4}"` afin de récupérer l’enregistrement dont l’id est 4, dans la collection spécifiée dans le champ **Collection**.

**Remarque :**
À la différence des instructions de requête requises dans le logiciel client MongoDB, la requête fait ici référence au contenu dans **find()**. Ainsi la requête est `{id:4}` alors qu’elle devrait être `db.blog.find({id:4})` dans le client MongoDB.

### Aggregation stages

Créez un pipeline d’agrégation en ajoutant les étapes à travers lesquelles vous souhaitez que les documents passent, pour obtenir des résultats agrégés à partir de ces documents. Cette table est disponible uniquement lorsque vous avez sélectionné **Aggregation pipeline query** dans la liste **Query type**.

Une seule étape est autorisée par ligne, dans la table **Aggregation stages**. Toutes les étapes sont exécutées une par une, dans l’ordre dans lequel vous les avez placées dans la table.

Par exemple, si vous souhaitez agréger les documents relatifs à vos clients, à l’aide des étapes `$match` et
$group, vous devez ajouter deux lignes à la table **Aggregation stages** et définir deux étapes, comme suit :

```
"$match : {status : 'A'}"
"$group : { _id : '$cust_id',
            total : {$sum : '$amount'}}"
```

Dans cette agrégation, les documents clients ayant un statut A sont sélectionnés. Parmi les clients sélectionnés, ceux utilisant le même ID client sont groupés et la somme des valeurs des champs *amount* du même client est effectuée.

Pour une liste complète des étapes que vous pouvez utiliser et les opérateurs associés, consultez Aggregation pipeline operators (en anglais).

Pour plus d’informations concernant le pipeline d’agrégation MongoDB, consultez Aggregation pipeline (en anglais).

**Mapping**

Chaque colonne du schéma défini pour ce composant représente un champ des documents à lire. Dans cette table, vous devez spécifier les nœuds parents de ces champs, s’il y en a.

Par exemple, dans le document se présentant comme suit :

```
{  
  _id: ObjectId("5099803df3f4948bd2f98391"),
  person: { first: "Joe", last: "Walker" }
}
```

Les champs *first* et *last* ont un nœud père *person* mais le champ *_id* ne contient aucun nœud père. Cela fait, la table **Mapping** doit ressembler à ceci :

<table>
<thead>
<tr>
<th>Column</th>
<th>Parent node path</th>
</tr>
</thead>
<tbody>
<tr>
<td>_id</td>
<td></td>
</tr>
<tr>
<td>first</td>
<td>&quot;person&quot;</td>
</tr>
<tr>
<td>last</td>
<td>&quot;person&quot;</td>
</tr>
</tbody>
</table>

**Sort by**

Spécifiez la colonne et choisissez l’ordre pour l’opération de tri.

Ce champ est disponible uniquement lorsque vous avez sélectionné **Find query** dans la liste **Query type**.

**Limit**

Saisissez le nombre maximal de documents à récupérer.

Ce champ est disponible uniquement lorsque vous avez sélectionné **Find query** dans la liste **Query type**.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.
**No query timeout**

Cochez cette case pour empêcher les serveurs MongoDB de fermer les curseurs inactifs après 10 minutes d'inactivité de ces curseurs. Dans cette situation, un curseur inactif reste ouvert jusqu'à ce que les résultats de ce curseur soient épuisés ou que vous fermez manuellement le curseur à l'aide de la méthode `cursor.close()`.

Un curseur, dans MongoDB, est un pointeur vers l'ensemble de résultats de la requête. Par défaut, c'est-à-dire lorsque la case est décochée, un serveur MongoDB ferme automatiquement les curseurs inactifs après une période d'inactivité donnée, pour éviter les utilisations excessives de mémoire. Pour plus d'informations concernant les curseurs MongoDB, consultez [https://docs.mongodb.org/manual/core/cursors/](https://docs.mongodb.org/manual/core/cursors/) (en anglais).

**Enable external sort**

Puisque les étapes des pipelines d'agrégation ont une limite d'utilisation de mémoire maximum (100 mégaoctets) et qu'une étape dépassant cette limite produit des erreurs, lors du traitement de jeux de données volumineux, cochez cette case pour éviter que les étapes d'agrégation dépassent cette limite.

Pour plus d’informations concernant ce tri externe, consultez [Large sort operation with external sort](https://docs.mongodb.org/manual/core/external-sort/) (en anglais).

### Global Variables

**Global Variables**

- **NB_LINE**: nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.
- **ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

**Règle d’utilisation**

Le composant `tMongoDBInput` est un composant de début et vous permet de récupérer des documents d’une collection, dans la base de données MongoDB et de
Scénario : Récupérer des données d’une collection via des requêtes avancées

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Dans ce scénario, des requêtes avancées de MongoDB sont utilisées pour récupérer le post écrit par Anderson.

Ces posts sont contenus dans la collection blog de la base de données MongoDB talend :

```javascript
> db.blog.find()
{ "id" : ObjectId("5226d25e675f26b4a2876888"), "id" : 1, "post" : { "author" : "Andy", "title" : "Open Source Outlook", "keywords" : "Open Source,Talend", "contents" : "Talend, the leader of the open source world..." } }
{ "id" : ObjectId("5226d25e675f26b4a2876889"), "id" : 2, "post" : { "author" : "Andy", "title" : "Data Integration Overview", "keywords" : "Data Integration,Talend", "contents" : "Talend, the leading player in the DI field..." } }
{ "id" : ObjectId("5226d25e675f26b4a287688a"), "id" : 3, "post" : { "author" : "Anderson", "title" : "ELT Overview", "keywords" : "ELT,Talend", "contents" : "Talend, the big name in the ELT circle..." } }
{ "id" : ObjectId("5226d25e675f26b4a2876888b"), "id" : 4, "post" : { "author" : "Andy", "title" : "Big Data Bang", "keywords" : "Big Data,Talend", "contents" : "Talend, the driving force for Big Data applications..." } }
```

Pour insérer des données dans la base de données, consultez Scénario : Créer une collection et écrire des données dans celle-ci à la page 2498.

Relier les composants

Procédure

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un tMongoDBConnection, un tMongoDBClose, un tMongoDBInput et un tLogRow.
2. Reliez le tMongoDBConnection au tMongoDBInput à l’aide d’un lien OnSubjobOk.
4. Reliez le tMongoDBInput au tLogRow à l’aide d’un lien Row > Main.
Configurer les composants

Procédure

1. Double-cliquez sur le composant **tMongoDBConnection** pour ouvrir sa vue **Basic settings**.

2. Dans la liste **DB Version**, sélectionnez la version de MongoDB que vous utilisez.
3. Dans les champs **Server** et **Port**, saisissez les détails de la connexion.
4. Dans le champ **Database**, saisissez le nom de la base de données MongoDB.
5. Double-cliquez sur le **tMongoDBInput** pour ouvrir sa vue **Basic settings**.
6. Cochez la case **Use existing connection**.

7. Dans le champ **Collection**, saisissez le nom de la collection, `blog`.

8. Cliquez sur le bouton `[...]` à côté du champ **Edit schema** pour ouvrir l’éditeur du schéma.

9. Cliquez sur le bouton `+` pour ajouter cinq colonnes, nommées respectivement `id` (de type **Integer**), `author`, `title`, `keywords` et `contents`, de type **String**.

10. Cliquez sur **OK** pour fermer l’éditeur.

11. Les colonnes apparaissent dans la partie gauche de la zone **Mapping**.

12. Pour les colonnes `author`, `title`, `keywords` et `contents`, saisissez le nœud parent `post`, afin que les données soient récupérées de leurs positions.
13. Dans le champ **Query**, saisissez la requête avancée afin de récupérer les posts dont l’auteur est Anderson :

```
"{post.author : 'Anderson'}"
```

Cette instruction nécessite que le sous-nœud de *post*, le nœud *author*, doit avoir la valeur "Anderson".

14. Double-cliquez sur le tLogRow pour ouvrir sa vue **Basic settings**.

Sélectionnez l’option **Table (print values in cells of a table)** pour un affichage optimal des résultats.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.

<table>
<thead>
<tr>
<th>id</th>
<th>author</th>
<th>title</th>
<th>keywords</th>
<th>contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Anderson</td>
<td>ELT Overview</td>
<td>ELT, Talend</td>
<td>Talend. the big name in the ELT circle...</td>
</tr>
</tbody>
</table>

Comme affiché ci-dessus, le post d’Anderson est récupéré.

**Scénario associé**

Pour un scénario associé, consultez :

- **Scénario : Créer une collection et écrire des données dans celle-ci** à la page 2498.
- **Scénario : Utiliser les fonctions MongoDB pour créer une collection et y écrire des données** à la page 2515.
tMongoDBOutput

Ce composant exécute une action définie sur une collection dans la base de données MongoDB.
Le tMongoDBOutput insère, met à jour, insère et met à jour ou supprime des documents dans une collection d’une base de données MongoDB, à partir du flux entrant du composant précédent dans le Job.

Propriétés du tMongoDBOutput Standard

Ces propriétés sont utilisées pour configurer le tMongoDBOutput s’exécutant dans le framework de Jobs Standard.
Le composant tMongoDBOutput Standard appartient aux familles Big Data et Databases.
Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Use existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Version</td>
<td>Sélectionnez dans la liste la version de la base de données que vous utilisez. Cette option est disponible lorsque la case Use existing connection est décochée.</td>
</tr>
<tr>
<td>Use replica set address</td>
<td>Cochez cette case pour afficher la table Replica address. Dans la table Replica address, vous pouvez configurer différents serveurs de la base de données MongoDB pour le failover. Cette option est disponible lorsque la case Use existing connection n’est pas cochée.</td>
</tr>
<tr>
<td>Server et Port</td>
<td>Adresse IP et port d’écoute du serveur de la base de données. Ces champs sont disponibles lorsque les cases Use existing connection et Use replica set address ne sont pas cochées.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Use SSL connection</td>
<td>Cochez cette case pour activer la connexion chiffrée SSL ou TLS. Utilisez le composant tSetKeystore dans le même Job afin de spécifier les informations de chiffrage. Pour plus d’informations concernant le tSetKeystore, consultez tSetKeystore à la page 3745.</td>
</tr>
</tbody>
</table>
Notez que la connexion SSL est disponible uniquement à partir de la version 2.4 de MongoDB.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set write concern</strong></td>
<td>Cochez cette case pour configurer le niveau d’accusé de réception requis par MongoDB pour les opérations d'écriture. Sélectionnez le niveau de cette opération. Pour plus d’informations, consultez la documentation relative à MongoDB à l’adresse suivante : <a href="http://docs.mongodb.org/manual/core/write-concern/">http://docs.mongodb.org/manual/core/write-concern/</a> (en anglais).</td>
</tr>
<tr>
<td><strong>Bulk write</strong></td>
<td>Cochez cette case pour insérer, mettre à jour ou supprimer des données en masse. Notez que cette fonctionnalité est disponible uniquement lorsque la version de MongoDB que vous utilisez est 2.6 ou supérieure. Vous devez sélectionner <strong>Ordered</strong> ou <strong>Unordered</strong> afin de définir comment la base de données MongoDB traite les données envoyées par le Studio. Si vous sélectionnez <strong>Ordered</strong>, MongoDB traite les données en séquence. Si vous sélectionnez <strong>Unordered</strong>, MongoDB optimise les opérations d'écriture en masse sans conserver l'ordre dans lequel les opérations individuelles ont été insérées dans l'écriture en masse. Dans le champ <strong>Bulk write size</strong>, saisissez la taille de chaque groupe de requêtes à traiter par MongoDB. Dans la documentation de MongoDB, certaines restrictions et certains comportements attendus, notamment la taille mentionnée ici, sont expliqués. Pour plus d’informations, consultez <a href="http://docs.mongodb.org/manual/core/bulk-write-operations/">http://docs.mongodb.org/manual/core/bulk-write-operations/</a> (en anglais).</td>
</tr>
<tr>
<td><strong>Required authentication</strong></td>
<td>Cochez cette case pour activer l’authentification à la base de données. Parmi les mécanismes listés dans la liste déroulante <strong>Authentication mechanism</strong>, le mécanisme <strong>NEGOTIATE</strong> est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez. Pour plus d’informations sur les autres mécanismes de la liste, consultez <strong>MongoDB Authentication</strong> (en anglais) dans la documentation MongoDB.</td>
</tr>
<tr>
<td><strong>Set Authentication database</strong></td>
<td>Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ <strong>Authentication database</strong> qui s’affiche. Pour plus d’informations sur la base de données d’authentification MongoDB, consultez <strong>User Authentication database</strong> (en anglais).</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur à la base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. 
Ces champs sont disponibles lorsque la case Required authentication est cochée.  
Si le système de sécurité sélectionné dans la liste Authentication mechanism est Kerberos, saisissez les informations dans les champs suivants User principal, Realm et KDC server et non dans les champs Username et Password. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection</td>
<td>Nom de la collection dans la base de données MongoDB.</td>
</tr>
<tr>
<td>Drop collection if exist</td>
<td>Cochez cette case afin de supprimer la collection si elle existe déjà.</td>
</tr>
</tbody>
</table>
| Action on data | Les opérations suivantes sont disponibles :  
• **Insert** : insère des documents.  
• **Set** : modifie les champs existants d’un document existant et écrit à la suite un champ s’il n’existe pas dans ce document.  
Si vous devez appliquer cette action sur tous les documents dans le collection à utiliser, cochez la case **Update all document** affichée. Sinon, seul le premier document est mis à jour.  
• **Update** : remplace les documents existants par les données d’entrée mais conserve l’ID technique de ces documents.  
• **Upsert** : insère un document s’il n’existe pas, sinon applique les mêmes règles que l’action **Update**.  
• **Upsert with set** : insère un document s’il n’existe pas, sinon applique les mêmes règles que l’action **Set**.  
Si vous devez appliquer cette action sur tous les documents de la collection à utiliser, cochez la case **Update all document** qui s’affiche. Sinon, seul le premier document est mis à jour.  
• **Delete** : supprime les documents. |
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• **View schema** : sélectionnez cette option afin de voir le schéma. |
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job.

### Built-In

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository

**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Mapping

Chaque colonne du schéma défini pour ce composant représente un champ des documents à lire. Dans cette table, vous devez spécifier les nœuds parents de ces champs, s’il y en a.

Par exemple, dans le document se présentant comme suit :

```json
{
    _id: ObjectId("5099803df3f4948bd2f98391"),
    person: { first: "Joe", last: "Walker" }
}
```

Les champs **first** et **last** ont un nœud père **person** mais le champ **_id** ne contient aucun nœud père. Cela fait, la table **Mapping** doit ressembler à ceci :

<table>
<thead>
<tr>
<th>Column</th>
<th>Parent node path</th>
</tr>
</thead>
<tbody>
<tr>
<td>_id</td>
<td></td>
</tr>
<tr>
<td>first</td>
<td>&quot;person&quot;</td>
</tr>
<tr>
<td>last</td>
<td>&quot;person&quot;</td>
</tr>
</tbody>
</table>

Ce tableau est indisponible lorsque la case **Generate JSON Document** est cochée dans l’onglet **Advanced settings**.
**Die on error**

Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

**Advanced settings**

**Generate JSON Document**

Cochez cette case pour la configuration JSON.

- **Configure JSON Tree** : Cliquez sur le bouton [...] pour ouvrir l’interface de configuration de l’arborescence JSON. Pour plus d’informations, consultez Configurer une arborescence JSON à la page 4287.
- **Group by** : Cliquez sur le bouton [+] afin d’ajouter des lignes et sélectionner les colonnes d’entrée pour grouper les enregistrements.
- **Remove root node** : Cochez cette case pour supprimer le nœud racine.
- **Data node** et **Query node** (disponibles pour les actions **Update** et **Upsert**) : Saisissez le nom du nœud de données et du nœud de requêtes configurés dans l’arborescence JSON.

⚠️ **Avertissement** : Ces nœuds sont obligatoires pour les actions **Update** et **Upsert**. Ils permettent d’activer les actions **Update** et **Upsert** mais ne seront pas stockés dans la base de données.

**No query timeout**

Cochez cette case pour empêcher les serveurs MongoDB de fermer les curseurs inactifs après 10 minutes d’inactivité de ces curseurs. Dans cette situation, un curseur inactif reste ouvert jusqu’à ce que les résultats de ce curseur soient épuisés ou que vous fermez manuellement le curseur à l’aide de la méthode `cursor.close()`.

Un curseur, dans MongoDB, est un pointeur vers l’ensemble de résultats de la requête. Par défaut, c’est-à-dire lorsque la case est décochée, un serveur MongoDB ferme automatiquement les curseurs inactifs après une période d’inactivité donnée, pour éviter les utilisations excessives de mémoire. Pour plus d’informations concernant les curseurs MongoDB, consultez https://docs.mongodb.org/manual/core/cursors/ (en anglais).

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

- **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères.
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Le <strong>tMongoDBOutput</strong> exécute l’action définie sur la collection dans la base de données MongoDB, à partir du flux entrant depuis le composant précédent dans le Job.</th>
</tr>
</thead>
</table>

**Limitation**

<table>
<thead>
<tr>
<th>Remarque :</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Le paramètre “multi”, qui vous permet de mettre à jour plusieurs documents en même temps, n’est pas supporté. Par conséquent, si deux documents ont la même clé, le premier est toujours mis à jour, mais le second ne l’est jamais.</td>
</tr>
<tr>
<td>• Pour l’opération de mise à jour, la clé ne peut être un tableau (array) JSON.</td>
</tr>
</tbody>
</table>

**Scénario : Créer une collection et écrire des données dans celle-ci**

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario crée la collection *blog* et écrit des données dans cette collection.

**Relier les composants**

**Procédure**

1. Déposez de la **Palette** dans l’espace de modélisation graphique un composant **tMongoDBConnection**, un **tFixedFlowInput**, un **tMongoDBOutput**, un **tMongoDBClose**, un **tMongoDBInput** et un **tLogRow**.

2. Renommez le **tFixedFlowInput** *blog_post_data*, le **tMongoDBOutput** *write_data_to_collection*, le **tMongoDBInput** *read_data_from_collection* et le **tLogRow** *show_data_from_collection*.

3. Reliez le **tMongoDBConnection** au **tFixedFlowInput** à l’aide d’un lien **OnSubjobOk**.

4. Reliez le **tFixedFlowInput** au **tMongoDBOutput** à l’aide d’un lien **Row > Main**.

5. Connectez le **tFixedFlowInput** au **tMongoDBInput** avec un lien **OnSubjobOk**.

6. Reliez le **tMongoDBInput** au **tMongoDBClose** à l’aide d’un lien **OnSubjobOk**.

7. Reliez le **tMongoDBInput** au **tLogRow** à l’aide d’un lien **Row > Main**.
Configurer les composants

Procédure

1. Double-cliquez sur le tMongoDBConnection pour ouvrir sa vue Basic settings.

2. Dans la liste DB Version, sélectionnez la version de MongoDB que vous utilisez.

3. Dans les champs Server et Port, saisissez les informations de connexion.
   Dans le champ Database, saisissez le nom de la base de données MongoDB.

Sélectionnez Use Inline Content (delimited file) dans la zone Mode.
Dans le champ Content, saisissez les données à écrire dans la base de données MongoDB, par exemple :

```
1;Andy;Open Source Outlook;Open Source,Talend,Talend, the leader of the open source world...
3;Andy;ELT Overview;ELT,Talend,Talend, the big name in the ELT circle...
2;Andy;Data Integration Overview;Data Integration,Talend,Talend, the leading player in the DI field...
```

5. Double-cliquez sur le tMongoDBOutput pour ouvrir sa vue Basic settings.

Cochez les cases Use existing connection et Drop collection if exist.
Dans le champ **Collection**, saisissez le nom de la collection, par exemple *blog*.

6. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur du schéma.

7. Cliquez cinq fois sur le bouton [*+*] pour ajouter cinq colonnes, à droite. Nommez-les respectivement *id*, *author*, *title*, *keywords* et *contents*. Définissez leur type à *Integer* pour la première colonne et à *String* pour les autres.

Cliquez sur le bouton ![copier](image.png) pour copier toutes les colonnes vers la table d’entrée.
Cliquez sur **OK** pour fermer l’éditeur.

8. Les colonnes apparaissent dans la partie gauche de la zone **Mapping**.
Dans la colonne **Parent node path** des lignes *author*, *title*, *keywords* et *contents*, saisissez le nœud père, *post*. Ces nœuds se trouvent à présent sous le nœud *post* dans la collection *MongoDB*.

9. Double-cliquez sur le composant **tMongoDBInput** pour ouvrir sa vue **Basic settings**.
Cochez la case **Use existing connection**.
Dans le champ **Collection**, saisissez le nom de la collection, *blog*.

10. Cliquez sur le bouton `[...]` à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

Cliquez sur **OK** pour fermer l'éditeur.

12. Les colonnes apparaissent dans la partie gauche de la zone **Mapping**.
Dans la colonne **Parent node path** des lignes **author**, **title**, **keywords** et **contents**, saisissez le nœud père, **post**, afin que ces données puissent être récupérées à partir de leur position.

13. Dans la zone **Sort by**, cliquez sur le bouton [+] pour ajouter une ligne et saisissez id sous **Column**. Sélectionnez **asc** dans la colonne **Order asc or desc?** à droite de la colonne id. Ainsi, les documents récupérés apparaîtront en ordre ascendant.

**Exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.

3. Passez à la base de données *talend* et lisez les données de la collection *blog*, dans l’invite de commande du client MongoDB. Vous pouvez constater que **author**, **title**, **keywords** et **contents** se trouvent sous le nœud **post**. Les enregistrements sont stockés dans le même ordre que dans la source des données.

**Scénario 2 : Effectuer une opération d’upsert sur des enregistrements dans une collection**

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario effectue une opération d’upsert sur la collection *blog*, car l’auteur d’un enregistrement existant a été modifié et qu’un nouvel enregistrement a été ajouté. Avant l’upsert, la collection *blog* se présente comme suit :

1; Andy; Open Source Outlook; Open Source, Talend; Talend, the leader of the open source world...
2; Andy; Data Integration Overview; Data Integration, Talend; Talend, the leading player in the DI field...
Relier les composants

Procédure
1. Déposez un composant `tMongoDBConnection`, un `tFixedFlowInput`, un `tMongoDBOutput`, un `tMongoDBClose`, un `tMongoDBInput` et un `tLogRow` de la Palette dans l’espace de modélisation graphique.
2. Renommez le `tFixedFlowInput` en `blog_post_data`, le `tMongoDBOutput` en `write_data_to_collection`, le `tMongoDBInput` en `read_data_from_collection` et le `tLogRow` en `show_data_from_collection`.
4. Reliez le `tFixedFlowInput` au `tMongoDBOutput` à l’aide d’un lien `Row > Main`.
7. Reliez le `tMongoDBInput` au `tLogRow` à l’aide d’un lien `Row > Main`.

Configurer les composants

Procédure
1. Double-cliquez sur le composant `tMongoDBConnection` pour ouvrir sa vue `Basic settings`.
2. Dans la liste **DB Version**, sélectionnez la version de MongoDB que vous utilisez.

3. Dans les champs **Server** et **Port**, saisissez les informations de connexion. Dans le champ **Database**, saisissez le nom de la base de données MongoDB.

4. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

Sélectionnez l’option **Use Inline Content (delimited file)** dans la zone **Mode**.

Dans le champ **Content**, saisissez les données pour effectuer l’upsert dans la base de données MongoDB, par exemple :

```
1;Andy;Open Source Outlook;Open Source,Talend;Talend, the leader of the open source world...
2;Andy;Data Integration Overview;Data Integration,Talend;Talend, the leading player in the DI field...
3;Anderson;ELT Overview;ELT,Talend;Talend, the big name in the ELT circle...
4;Andy;Big Data Bang;Big Data,Talend;Talend, the driving force for Big Data applications...
```
Comme affiché ci-dessus, l'auteur du troisième enregistrement a été modifié et le quatrième enregistrement est nouveau.

5. Double-cliquez sur le tMongoDBOutput pour ouvrir sa vue Basic settings.

6. Cochez les cases Use existing connection et Die on error.

7. Dans le champ Collection, saisissez le nom de la collection, blog.

8. Sélectionnez Upsert dans la liste Action on data.

6. Cliquez sur le bouton [...] à côté de Edit schema pour ouvrir l’éditeur de schéma.


8. Cliquez sur le bouton pour copier toutes les colonnes vers la table d’entrée.

Dans la vue Advanced settings, cochez la case Generate JSON Document.

8. Cochez la case Remove root node.

Dans les champs Data node et Query node, saisissez “data” et “query”.

Cliquez sur OK pour fermer l’éditeur.
9. Cliquez sur le bouton [...] à côté de **Configure JSON Tree** pour ouvrir l’interface de configuration.

10. Cliquez-droit sur le nœud `rootTag` et sélectionnez **Add Sub-element** dans le menu contextuel. Dans la boîte de dialogue qui s’ouvre, saisissez `data`. 
Cliquez sur OK pour fermer la fenêtre.
Répétez cette opération et saisissez query.
Cliquez-droit sur le nœud data et sélectionnez Set As Loop Element dans le menu contextuel.

Avertissement :
Ces nœuds sont obligatoires pour les actions Update et Upsert. Ils permettent d'activer les actions Update et Upsert mais ne seront pas stockés dans la base de données.

11. Sélectionnez toutes les colonnes sous la liste Schema list et déposez-les dans le nœud data.
Dans la fenêtre qui s'ouvre, sélectionnez Create as sub-element of target node.

Cliquez sur OK pour fermer la fenêtre.
Répétez l'opération pour déposer la colonne id de la liste Schema list sous le nœud Query.

12. Cliquez-droit sur le nœud id sous data et sélectionnez Add Attribute dans le menu contextuel.
Dans la boîte de dialogue qui s'ouvre, saisissez type comme nom d'attribut :

Cliquez sur OK pour fermer la fenêtre.
Cliquez-droit sur le nœud @type sous id et sélectionnez Set A Fix Value dans le menu contextuel.
Dans la boîte de dialogue qui s’ouvre, saisissez integer comme valeur d’attribut, afin de vous assurer que les valeurs de id sont stockées en tant qu’entiers dans la base de données.

Cliquez sur OK pour fermer la fenêtre.
Répétez l’opération afin de configurer cet attribut pour le nœud id sous Query.
Cliquez sur OK pour fermer l’interface de configuration de l’arborescence JSON.


Cochez la case Use existing connection.
Dans le champ Collection, saisissez le nom de la collection, blog.
Cliquez sur le bouton [...] à côté du champ Edit schema pour ouvrir l’éditeur de schéma.
Cliquez cinq fois sur le bouton [+] pour ajouter cinq colonnes. Nommez-les id, author, title, keywords et contents et configurez le type de la colonne id à Integer et des autres colonnes à String.

Cliquez sur OK pour fermer l’éditeur.

Les colonnes apparaissent dans la partie gauche de la zone Mapping area.

Saisissez le nœud père post pour les colonnes author, title, keywords et contents, afin que les données puissent être récupérées de leurs positions correctes.


Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour un meilleur affichage.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl+S pour sauvegarder le Job.
2. Appuyez sur F6 pour exécuter le Job.

Comme affiché ci-dessus, l’auteur du troisième enregistrement est mis à jour et le quatrième enregistrement est inséré.
tMongoDBRow

Ce composant exécute les commandes et les fonctions de la base de données MongoDB.

Propriétés du tMongoDBRow Standard

Ces propriétés sont utilisées pour configurer le tMongoDBRow s'exécutant dans le framework de Jobs Standard.

Le composant tMongoDBRow Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez dans la liste la version de la base de données que vous utilisez. Cette option est disponible lorsque la case Use existing connection est décochée.</td>
</tr>
<tr>
<td>Use replica set address</td>
<td>Cochez cette case pour afficher la table Replica address.</td>
</tr>
<tr>
<td></td>
<td>Dans la table Replica address, vous pouvez configurer différents serveurs de la base de données MongoDB pour le failover.</td>
</tr>
<tr>
<td></td>
<td>Cette option est disponible lorsque la case Use existing connection n'est pas cochée.</td>
</tr>
<tr>
<td>Server et Port</td>
<td>Adresse IP et port d'écoute du serveur de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Ces champs sont disponibles lorsque les cases Use existing connection et Use replica set address ne sont pas cochées.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Use SSL connection</td>
<td>Cochez cette case pour activer la connexion chiffrée SSL ou TLS.</td>
</tr>
<tr>
<td></td>
<td>Utilisez le composant tSetKeystore dans le même Job afin de spécifier les informations de chiffrement.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d'informations concernant le tSetKeystore, consultez tSetKeystore à la page 3745.</td>
</tr>
<tr>
<td></td>
<td>Notez que la connexion SSL est disponible uniquement à partir de la version 2.4 de MongoDB.</td>
</tr>
</tbody>
</table>
**Required authentication**

Cochez cette case pour activer l’authentification à la base de données.

Parmi les mécanismes listés dans la liste déroulante **Authentication mechanism**, le mécanisme **NEGOTIATE** est recommandé si vous n’utilisez pas Kerberos, car il sélectionne automatiquement le mécanisme d’authentification le plus adapté à la version de MongoDB que vous utilisez.

Pour plus d’informations sur les autres mécanismes de la liste, consultez MongoDB Authentication (en anglais) dans la documentation MongoDB.

---

**Set Authentication database**

Si le nom d’utilisateur à utiliser pour se connecter à MongoDB a été créé dans une base de données d’authentification MongoDB spécifique, cochez cette case pour saisir le nom de la base de données en question dans le champ **Authentication database** qui s’affiche.

Pour plus d’informations sur la base de données d’authentification MongoDB, consultez User Authentication database (en anglais).

---

**Username et Password**

Informations d’authentification de l’utilisateur à la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Ces champs sont disponibles lorsque la case **Required authentication** est cochée.

Si le système de sécurité sélectionné dans la liste **Authentication mechanism** est Kerberos, saisissez les informations dans les champs suivants **User principal**, **Realm** et **KDC server** et non dans les champs **Username** et **Password**.

---

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez
propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent.

### Execute command

Cochez cette case afin de saisir des commandes MongoDB dans le champ Command pour l'exécution.

- **Command** : dans ce champ, saisissez la commande à exécuter, si cette commande contient une seule variable.

  Par exemple, si vous devez construire la commande suivante :

  ```
  {"isMaster": 1}
  ```

  Saisissez simplement `isMaster` entre guillements.

- **Construct command from keys and values** : si la commande à exécuter contient différentes variables, cochez cette case et, dans la table **Command keys and values**, ajoutez les variables et leurs valeurs à utiliser.

  Par exemple, si vous devez construire la commande suivante :

  ```
  { renameCollection : "<source_namespace>",
    to : "<target_namespace>",
    dropTarget : < true | false > }
  ```

  Vous devez ajouter trois lignes à la table **Command keys and values** et saisir une paire variable-valeur pour chaque ligne, entre guillemets :

  ```
  "renameCollection" "old_name"
  "to" "new_name"
  "dropTarget" "false"
  ```

- **Construct command from a JSON string** : si vous souhaitez saisir directement la commande à utiliser, cochez cette case et saisissez cette commande dans le champ **JSON string command** affiché.

  Une commande est autorisée par composant tMongoDBRow.

  Par exemple :

  ```
  "{createIndexes: 'restaurants',
   indexes : [{key :
    {restaurant_id: 1}, name:
    'id_index_2', unique: true}]}"
  ```

  Notez que vous devez utiliser des guillemets simples pour entourer les valeurs String utilisées dans la
commande et des guillemets doubles pour entourer la commande elle-même.

Pour plus d’informations concernant les commandes MongoDB que vous pouvez utiliser dans ce champ, consultez https://docs.mongodb.org/manual/reference/command/ (en anglais).

### Function
Saisissez les fonctions MongoDB dans le champ **Function** pour l’exécution.

Ce champ est indisponible lorsque la case **Execute command** est cochée.

### Parameters value
Cliquez sur le bouton [+] pour ajouter des lignes, puis configurez la valeur des paramètres sous forme de valeurs variables ou constantes, par exemple `row1.author` ou ‘Andy’. La valeur des paramètres correspond aux paramètres définis dans le champ **Function**, dans le même ordre.

Cette option est indisponible lorsque la case **Execute command** est cochée.

### Die on error
Cette case est décochée par défaut, afin d’ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

2514
Utilisation

| Règle d’utilisation | Le composant tMongoDBRow vous permet de manipuler la base de données MongoDB à l’aide des commandes et des fonctions MongoDB. |

Scénario : Utiliser les fonctions MongoDB pour créer une collection et y écrire des données

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario crée la collection blog et y écrit des données à l’aide des fonctions MongoDB.

Relier les composants

Procédure

1. Déposez, de la Palette dans l’espace de modélisation graphique, un composant tMongoDBConnection, un tFixedFlowInput, un tMongoDBRow, un tMongoDBClose, un tMongoDBInput et un tLogRow.
2. Renommez le tFixedFlowInput en blog_post_data, le tMongoDBRow en write_data_to_collection, le tMongoDBInput en read_data_from_collection et le tLogRow en show_data_from_collection.
4. Reliez le tFixedFlowInput au tMongoDBRow à l’aide d’un lien Row > Main.
5. Connectez le tFixedFlowInput au tMongoDBInput en utilisant un lien OnSubjobOk.
7. Reliez le tMongoDBInput au composant tLogRow à l’aide d’un lien Row > Main.

Configurer les composants

Procédure

1. Double-cliquez sur le tMongoDBConnection pour ouvrir sa vue Basic settings.
2. Dans la liste **DB Version**, sélectionnez la version de MongoDB que vous utilisez.

3. Dans les champs **Server** et **Port**, saisissez le nom du serveur et le numéro du port, respectivement. Dans le champ **Database**, saisissez le nom de la base de données MongoDB.

4. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

Sélectionnez l'option **Use Inline Content (delimited file)** dans la zone **Mode**.

Dans le champ **Content**, saisissez les données à écrire dans la base de données, par exemple :

Andy;Open Source Outlook;Open Source,Talend;Talend, the leader of the open source world...
Andy;Data Integration Overview;Data Integration,Talend;Talend, the leading player in the DI field...
Andy;ELT Overview;ELT,Talend;Talend, the big name in the ELT circle...

5. Double-cliquez sur le **tMongoDBRow** pour ouvrir sa vue **Basic settings**.
Cochez la case **Use existing connection**.
Dans le champ **Function**, saisissez la fonction MongoDB pour créer la collection *blog* et y insérer des données :

```
"function(author,title,keywords,contents){
    return db.blog.save(
        {author:author,title:title,keywords:keywords,contents:contents}
    );
}
```

6. Cliquez sur le bouton [..] à côté de **Edit schema** pour ouvrir l'éditeur du schéma.
7. Cliquez quatre fois sur le bouton [+], pour ajouter quatre colonnes, à droite. Nommez-les respectivement author, title, keywords et contents, toutes de type String.

Cliquez sur le bouton pour copier toutes les colonnes vers la table d'entrée. Cliquez sur OK pour fermer l'éditeur.


10. Cochez la case Use existing connection.
    Dans le champ Collection, saisissez le nom de la collection, blog.
    Cliquez sur le bouton [...] à côté du champ Edit schema pour ouvrir l'éditeur du schéma.
Cliquez sur OK pour fermer l’éditeur.

12. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings.

Dans la zone Mode, sélectionnez Table (print values in cells of a table), pour un meilleur affichage.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl+S pour sauvegarder le Job.
2. Appuyez sur la touche F6 pour exécuter le Job.
tMSAXInput

Ce composant extrait des données d’un serveur MicrosoftAX en se basant sur une requête.

Propriétés du tMSAXInput Standard

Ces propriétés sont utilisées pour configurer le tMSAXInput s’exécutant dans le framework de Jobs Standard.
Le composant tMSAXInput Standard appartient à la famille Business.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version</th>
<th>Sélectionnez le type de connecteur métier à utiliser. Ce type peut être :</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Dynamics AX 4.0</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>Dynamics AX 2012</strong></td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong> : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
<td></td>
</tr>
<tr>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
<td></td>
</tr>
<tr>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
<td></td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remarque :</strong></td>
<td>.NET Business Connector Assembly Path</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Host</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Port</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>AOS Server Instance</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Domain</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Username et Password</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Company</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Language</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Configuration File</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Schema et Edit Schema</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Table Name</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Query</td>
</tr>
</tbody>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tMSAXOutput

Ce composant permet écrit des données dans un serveur MicrosoftAX.

Propriétés du tMSAXOutput Standard

Ces propriétés sont utilisées pour configurer le tMSAXOutput s’exécutant dans le framework de Jobs Standard.
Le composant tMSAXOutput Standard appartient à la famille Business.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Version

Sélectionnez le type de connecteur métier à utiliser. Ce type peut être :
- Dynamics AX 4.0
- Dynamics AX 2012

Advanced settings

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est</td>
</tr>
</tbody>
</table>
une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remarque :</strong></td>
<td>Ce composant est généralement utilisé en tant que composant de sortie. Un composant de début est nécessaire.</td>
</tr>
</tbody>
</table>

1. **Remarque :**
   - Type *Dynamics AX 2012* uniquement.
   - **.NET Business Connector Assembly Path**

2. **Remarque :**
   - Type *Dynamics AX 2012* uniquement.
   - **Host**

3. **Remarque :**
   - Type *Dynamics AX 2012* uniquement.
   - **Port**

4. **Remarque :**
   - Type *Dynamics AX 2012* uniquement.
   - **AOS Server Instance**

5. **Remarque :**
   - Type *Dynamics AX 2012* uniquement.
   - **Domain**

6. **Remarque :**
   - Type *Dynamics AX 2012* uniquement.
   - **Username et Password**

7. **Remarque :**
   - Type *Dynamics AX 2012* uniquement.
   - **Company**

8. **Remarque :**
   - Type *Dynamics AX 2012* uniquement.
   - **Language**

9. **Remarque :**
   - Type *Dynamics AX 2012* uniquement.
   - **Configuration File**

10. **Remarque :**
    - Type *Dynamics AX 2012* uniquement.
    - **Table Name**

11. **Remarque :**
    - Type *Dynamics AX 2012* uniquement.
    - **Action on data**

12. **Remarque :**
    - Type *Dynamics AX 2012* uniquement.
    - **Schema et Edit Schema**

13. **Remarque :**
    - Type *Dynamics AX 2012* uniquement.
    - **Die on error**

14. **Remarque :**
    - Type *Dynamics AX 2012* uniquement.
    - **Additional columns**
Scénario : Insérer des données dans une table définie sur le serveur Microsoft AX

Avertissement :
Avant de pouvoir utiliser ce composant, vérifiez que vous avez installé et lancé le serveur Microsoft AX correctement.

Ce scénario décrit un Job à deux composants qui utilise le composant tMSAXOutput afin d’insérer une ligne dans une table définie stockée sur un serveur Microsoft AX et modifie ensuite les valeurs dans l’une des colonnes insérée.

Configurer le Job

Procédure
1. Glissez le composant tFixedFlowInput ainsi que le tMSAXOutput de la Palette dans l’espace de modélisation.
2. Connectez les composants à l’aide d’un lien Row de type Main.

Configurer le tFixedFlowInput

Procédure
1. Double-cliquez sur le tFixedFlowInput afin d’afficher la vue Component et de définir ses propriétés.

2. Sélectionnez le mode Built-In dans le champ Schema et cliquez sur le bouton [...] du champ Edit schema afin d’afficher une boîte de dialogue qui vous permettra de définir le schéma d’entrée.
3. Cliquez sur le bouton [+] vert pour ajouter des colonnes dans le schéma d’entrée, trois dans cet exemple, name, city et street (nom, ville et rue).

4. Cliquez sur OK pour fermer la boîte de dialogue. Une nouvelle boîte de dialogue s’ouvre et vous demande si vous voulez propager les modifications, cliquez sur Yes (Oui). Les colonnes du schéma s’affichent automatiquement dans le tableau Values.

5. Cliquez sur la colonne Value et saisissez une valeur pour chaque colonne d’entrée.

**Configurer le tMSAXOutput**

**Procédure**

1. Double-cliquez sur le tMSAXOutput pour ouvrir la vue Component et définir ses propriétés.

2. Sélectionnez le mode Built-In dans le champ Property Type.

3. Dans le champ Host, saisissez l’adresse IP du serveur MicrosoftAX et, dans le champ Domain, saisissez le nom du domaine qui héberge le serveur MicrosoftAX.

4. Entrez votre nom d’utilisateur et votre mot de passe de connexion au serveur dans les champs correspondants, puis, dans le champ Table Name, saisissez le nom de la table dans laquelle vous souhaitez écrire des données.

5. Dans la liste Action on data, sélectionnez l’action que vous désirez exécuter, Insert dans cet exemple.

6. Cliquez sur Sync columns pour retrouver le schéma du composant précédent.

   Dans cet exemple, le but est de retrouver les trois colonnes d’entrée : name, city et street (nom, ville et rue) et d’écrire les données comprises dans les trois colonnes d’entrée du serveur MicrosoftAX sans effectuer de modification.

   Si nécessaire, cliquez sur le bouton [...] du champ Edit schema afin d’ouvrir une boîte de dialogue qui vous permettra de vérifier le schéma retourné.

7. Dans le tableau Additional columns, cliquez sur le bouton [+] afin d’ajouter une ligne dans laquelle vous pouvez définir les paramètres de la nouvelle colonne à ajouter à la ligne que vous souhaitez écrire dans la table ADDRESS.

8. Définissez un nom, un type de données, une position et une colonne de référence dans les colonnes correspondantes de la ligne ajoutée.

   Dans cet exemple, ajoutez une nouvelle colonne nommée "address" après la colonne "street".
9. Cliquez sur la colonne **Local expression** et appuyez sur **Ctrl+Espace** sur votre clavier pour ouvrir la liste des variables de contexte et sélectionnez `StringHandling.UPCASE(row2.city)+"-"+row2.street`. Cette expression écrira le nom de la ville avec la première lettre en majuscule, suivi du nom de la rue, pour former l’adresse du *Bryant park*. La colonne *address* de cet exemple contiendra donc la chaîne de caractères suivante : *New York-Midtown Manhattan*.

**Exécuter le Job**

**Procédure**

Sauvegardez votre Job et appuyez sur **F6** pour l’exécuter.

Le **tMSAXOutput** insère dans la table *ADDRESS* du serveur MicrosoftAX une ligne contenant les trois colonnes d’entrée *name*, *city* et *street*, en plus de la nouvelle colonne *address* qui contient à la fois le nom de la ville et le nom de la rue.

**Scénario 2 : Effacer des données d’une table précise sur le serveur MicrosoftAX.**

⚠️ **Avertissement :**

Avant de pouvoir utiliser ce composant, vérifiez que vous avez installé et lancé correctement le serveur MicrosoftAX.

Ce scénario décrit un Job à deux composants qui utilise le **tMSAXOutput** dans le but d’effacer toutes les lignes d’une table définie qui ne correspondent pas aux données contenues dans la colonne clé sur un serveur MicrosoftAX.

Dans cet exemple, le schéma d’entrée que vous utilisez est une colonne *address* (adresse) qui contient les données suivantes : *New York-Midtown Manhattan*. Vous allez effacer du serveur MicrosoftAX toutes les adresses qui ne sont pas identiques à celle-ci.

**Configurer le Job**

**Procédure**

1. Glissez les composants **tFixedFlowInput** et **tMSAXOutput** de la Palette dans l’espace de modélisation graphique.
2. Connectez les deux composants à l’aide d’un lien **Row** de type **Main**.
Configurer le tFixedFlowInput

**Procédure**

1. Double-cliquez sur le tFixedFlowInput pour afficher sa vue Component et définir ses propriétés.

   ![Image of tFixedFlowInput Component](image)

2. Sélectionnez le mode Built-In dans le champ Schema et cliquez sur le bouton [...] du champ Edit schema pour ouvrir une boîte de dialogue où vous pourrez définir le schéma.

3. Cliquez sur le bouton [+] et ajoutez les colonnes d'entrée du schéma, address dans cet exemple.

4. Cliquez sur OK pour fermer la boîte de dialogue. Les colonnes du schéma s'affichent automatiquement dans la liste Values.

5. Cliquez sur la colonne Value et saisissez une valeur pour la colonne d'entrée.

Configurer la connexion au serveur MicrosoftAX

**Procédure**

1. Double-cliquez sur le tMSAXOutput pour afficher sa vue Component et définir ses propriétés.

   ![Image of tMSAXOutput Component](image)

2. Sélectionnez le mode Built-In dans le champ Property Type.

3. Dans le champ Host, saisissez l'adresse IP du serveur MicrosoftAX.

4. Dans le champ Domain, entrez le nom du domaine qui héberge le serveur MicrosoftAX.

5. Saisissez votre nom d'utilisateur et votre mot de passe de connexion au serveur dans les champs correspondants.
6. Dans le champ **Table Name**, saisissez le nom de la table dans laquelle vous voulez supprimer des données, **ADDRESS** dans cet exemple.

### Définir l’action sur les données

**Procédure**

1. Dans la liste **Action on data**, sélectionnez l’action que vous souhaitez exécuter, **Delete** dans cet exemple.
2. Cliquez sur **Sync columns** pour retourner le schéma du composant précédent. Dans cet exemple, l’objectif est de récupérer la colonne d’entrée **address**.
3. Cliquez sur le bouton [...] du champ **Edit Schema** pour ouvrir la boîte de dialogue dans laquelle vous pouvez vérifier le schéma récupéré.

![Schema of tMSAXOutput_2](image)

4. Dans le schéma de sortie, cochez la case **Key** (clé) à côté du nom de la colonne puis cliquez sur **OK** pour valider les changements et fermer la boîte de dialogue.

**Remarque :**

Lorsque vous sélectionnez l’action sur les données **Delete**, vous devez définir la colonne de référence **Reference column** comme colonne clé afin que le tMSAXOutput supprime les lignes à partir de cette colonne clé.

5. Dans le tableau **Additional columns**, cliquez sur le bouton [+] pour ajouter une ligne et définir les paramètres que le composant utilisera comme base pour l’opération d’effacement.
6. Saisissez un nom (**name**), un opérateur (**operator**), un type de données (**data type**), une expression locale (**local expression**), une position (**position**) et une colonne de référence (**reference column**) dans les colonnes correspondantes à la ligne que vous venez d’ajouter.

Dans cet exemple, l’objectif est d’effacer de la table **ADDRESS** du serveur MicrosoftAX toutes les lignes dans lesquelles la colonne **Address** n’est pas identique à l’adresse dans la colonne clé **address**, New York-Midtown Manhattan.

**Remarque :**

Quand vous sélectionnez l’option **Delete**, vous devez toujours paramétrer **Position** à **Replace**. Sinon, les paramètres du tableau **Additional columns** ne seront pas pris en compte lors de l’exécution de votre Job.
**Exécuter le Job**

**Procédure**

Sauvegardez votre Job et appuyez sur **F6** pour l’exécuter.

Le **tMSAXOutput** supprime de la table *ADDRESS* du serveur MicrosoftAX toutes les lignes pour lesquelles les données de la colonne *address* ne sont pas identiques à celles de la colonne clé.
tMsgBox

Ce composant ouvre une boîte de dialogue contenant un bouton **OK**, nécessitant une action de la part de l'utilisateur.

Le tMsgBox est une pause graphique dans le cours du traitement.

**Propriétés du tMsgBox Standard**

Ces propriétés sont utilisées pour configurer le tMsgBox s’exécutant dans le framework de Jobs Standard.

Le composant tMsgBox Standard appartient à la famille Misc.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Title</th>
<th>Le texte saisi s’affiche dans la barre de titre de la boîte de dialogue.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buttons</td>
<td>Liste des boutons que vous souhaitez inclure dans la boîte de dialogue. Les combinaisons de bouton sont restreintes et ne peuvent être changées.</td>
</tr>
<tr>
<td></td>
<td>Le bouton Question fait apparaître la case <strong>Masquer la réponse</strong>, qui une fois cochée vous permet de masquer la réponse que vous saisissez dans la fenêtre pop-up qui s’affiche quand vous exécutez votre Job.</td>
</tr>
<tr>
<td>Icon</td>
<td>Icône de la barre de titre de la boîte de dialogue.</td>
</tr>
<tr>
<td>Message</td>
<td>Texte libre à afficher dans la boîte de dialogue. Le texte peut être dynamique (ex : reprendre un nom de fichier).</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

| Global Variables | RESULT : valeur de retour du composant. Cette variable est une variable After et retourne une chaîne de caractères. |
|                 | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Scénario : Test de type 'Hello world!'

Le scénario suivant représente un Job à un seul composant utilisé pour afficher le pid (l’identifiant de processus) à la place du message "Hello World!".

Construire le Job

Pourquoi et quand exécuter cette tâche

Le scénario suivant représente un Job à un seul composant utilisé pour afficher le pid (l’identifiant de processus) à la place du message "Hello World!".

Procédure

1. Cliquez et déposez un composant tMsgBox dans l’espace de modélisation.
2. Paramétrez les propriétés d’affichage du message :

   Le champ Title représente le titre de la boîte de message, vous pouvez utiliser n’importe quelle variable.

3. Dans le champ Message, saisissez le texte "Current date is: " entre guillemets doubles concaténé à l’aide d’un +, puis appuyez sur Ctrl+Espace afin d’afficher la liste d’auto-complétion, et sélectionnez la routine système TalendDate.getCurrentDate. Entourez-la de parenthèses.

Résultats

Le message affiche le texte défini précédemment et requiert une action de l’utilisateur pour disparaître et passer au composant suivant ou terminer le Job.

Après avoir cliqué sur le bouton OK, le log de la vue Run est mis à jour.

Voir également le Guide utilisateur du Studio Talend.

Exécuter le Job

Procédure

Cliquez sur l’onglet Run puis exécutez le Job.

Résultats

Le message affiche le texte défini précédemment et requiert une action de l’utilisateur pour disparaître et passer au composant suivant ou terminer le Job.

Après avoir cliqué sur le bouton OK, le log de la vue Run est mis à jour.

Voir également le Guide utilisateur du Studio Talend.
tMSSqlBulkExec

Ce composant permet un gain de performance pendant les opérations d'Insert dans une base de données Microsoft SQL Server.

Les composants tMSSqlOutputBulk et tMSSqlBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tMSSqlOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Propriétés du tMSSqlBulkExec Standard

Ces propriétés sont utilisées pour configurer le tMSSqlBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tMSSqlBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>JDBC Provider</td>
<td>Sélectionnez le fournisseur du pilote JDBC à utiliser, Microsoft (recommandé) ou Open source JTDS.</td>
</tr>
<tr>
<td></td>
<td>Lorsque vous utilisez ce composant avec la source de données dans Talend Runtime, vous devez utiliser le pilote Open source JTDS.</td>
</tr>
<tr>
<td></td>
<td>Notez que, quand Microsoft est sélectionné, vous devez télécharger le pilote Microsoft JDBC pour les serveurs SQL sur Microsoft Download Center (en anglais). Vous devez ensuite décompresser le fichier zip téléchargé, choisir un fichier .jar dans le dossier décompressé, selon votre version de JRE, renommer le fichier .jar</td>
</tr>
</tbody>
</table>
en `mssql-jdbc.jar` et l’installer manuellement. Pour plus d’informations relatives au choix du fichier .jar, consultez la configuration système requise sur Microsoft Download Center (en anglais).

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. 
Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
</tbody>
</table>
| Username et Password       | Informations d’authentification de l’utilisateur de base de données. 
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Table                      | Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit exister pour que l’opération d’Insert soit autorisée. |
| Action on table            | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée : 
**None** : n’effectuer aucune opération de table. 
**Drop and create the table** : supprimer la table puis en créer une nouvelle. 
**Create a table** : créer une table qui n’existe pas encore. 
**Create table if doesn’t exist** : créer la table si nécessaire. 
**Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. 
**Clear a table** : supprimer le contenu de la table. 
**Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| Schema et Edit Schema       | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs. |
**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

**Repository** : Le schéma existe déjà et il est stocké dans le *Repository*. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre *[Repository Content]*.

**Remote File Name**

Nom du fichier à charger.

**Avertissement** :

Le fichier est situé sur la machine spécifiée par l’URI dans le champ *Host* et doit être sur la même machine que le serveur de la base de données.

### Advanced settings

**Action**


**Fire Triggers**

Cochez cette case pour exécuter tout déclencheur d’insertion défini sur la table dans laquelle les données vont être chargées durant l’opération d’insertion de masse.
Cette propriété est disponible lorsque l’option Bulk insert est sélectionnée dans la liste déroulante Action.

<table>
<thead>
<tr>
<th>Utilisation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Règle d’utilisation</td>
<td>Ce composant est généralement utilisé avec un composant tMSSqlOutputBulk. Ensemble, ils offrent un gain de performance important pour l’alimentation d’une base de données MSSql.</td>
</tr>
<tr>
<td>Bulk insert &amp; Bulk update</td>
<td>Additional JDBC parameters</td>
</tr>
<tr>
<td></td>
<td>Fields terminated</td>
</tr>
<tr>
<td></td>
<td>Rows terminated</td>
</tr>
<tr>
<td></td>
<td>First row</td>
</tr>
<tr>
<td></td>
<td>Code page</td>
</tr>
<tr>
<td></td>
<td>Data file type</td>
</tr>
<tr>
<td></td>
<td>Output</td>
</tr>
<tr>
<td></td>
<td>tStatCatcher Statistics</td>
</tr>
<tr>
<td>Bcp query out</td>
<td>Fields terminated</td>
</tr>
<tr>
<td></td>
<td>Rows terminated</td>
</tr>
<tr>
<td></td>
<td>Data file type</td>
</tr>
<tr>
<td></td>
<td>Output</td>
</tr>
<tr>
<td></td>
<td>tStatCatcher Statistics</td>
</tr>
</tbody>
</table>
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données.
dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.


Scénarios associés

Pour un scénario associé au composant tMSSqlBulkExec, consultez les scénarios suivants :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du tMysqlOutputBulk.

- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec.
tMSSqlClose

Ce composant ferme une connexion à la base de données MSSql.

Propriétés du tMSSqlClose Standard

Ces propriétés sont utilisées pour configurer le tMSSqlClose s'exécutant dans le framework de Jobs Standard.

Le composant tMSSqlClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tMSSqlConnection dans la liste.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
Lorsqu'un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

**Scénario associé**

Aucun scénario n'est disponible pour la version Standard de ce composant.
tMSSqlColumnList

Ce composant liste les noms des colonnes d'une table MS SQL donnée.
Le tMSSqlColumnList effectue une opération d'itération dans toutes les colonnes d'une table donnée, grâce à une connexion MS SQL définie.

Propriétés du tMSSqlColumnList Standard

Ces propriétés sont utilisées pour configurer le tMSSqlColumnList s'exécutant dans le framework de Jobs Standard.
Le composant tMSSqlColumnList Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tMSSqlConnection dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Table name</td>
<td>Saisissez le nom de la table.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez la case afin de collecter les données de log au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_NAME</td>
<td>nom de la colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>COLUMN_DEFAULT</td>
<td>valeur par défaut de la colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>IS_NULLABLE</td>
<td>nullabilité de la colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne YES si la colonne autorise les valeurs NULL. Sinon, elle retourne NO.</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DATA_TYPE</td>
<td>Type de données de la colonne sur laquelle se fait l'itération. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>CHARACTER_MAXIMUM_LENGTH</td>
<td>Nombre maximum de caractères de la colonne sur laquelle se fait l'itération. Cette variable est une variable Flow et retourne une chaîne de caractères numériques pour les types de données binaires ou caractères et retourne NULL pour les autres types.</td>
</tr>
<tr>
<td>CHARACTER_OCTET_LENGTH</td>
<td>Longueur maximale en octets de la colonne sur laquelle se fait l'itération. Cette variable est une variable Flow et retourne une chaîne de caractères numériques pour les types de données binaires ou caractères et retourne NULL pour les autres types.</td>
</tr>
<tr>
<td>NUMERIC_PRECISION</td>
<td>Précision des données de la colonne sur laquelle se fait l'itération. Cette variable est une variable Flow et retourne une chaîne de caractères numériques pour les données numériques approximatives, exactes, les entiers, les données monétaires et retourne NULL pour les autres types de données.</td>
</tr>
<tr>
<td>NUMERIC_PRECISION_RADIX</td>
<td>Base (radix) de précision des données de la colonne sur laquelle s'effectue l'itération. Cette variable est une variable Flow et retourne une chaîne de caractères numériques pour les données numériques approximatives, exactes, les entiers, les données monétaires et retourne NULL pour les autres types de données.</td>
</tr>
<tr>
<td>NUMERIC_SCALE</td>
<td>L'échelle en chiffres autorisée sur la colonne sur laquelle l'itération s'effectue. Cette variable est une variable Flow et retourne une chaîne de caractères numériques pour les données numériques approximatives, exactes, les entiers, les données monétaires et retourne NULL pour les autres types de données.</td>
</tr>
<tr>
<td>DATETIME_PRECISION</td>
<td>Précision de l'heure et de la date en secondes fractionnelles de la colonne sur laquelle s'effectue l'itération. Cette variable est une variable Flow et retourne une chaîne de caractères numériques si le type de données est datetime ou smalldatetime. Sinon, retourne NULL.</td>
</tr>
<tr>
<td>COLUMN_KEY</td>
<td>Indicateur clé de la colonne sur laquelle se fait l'itération. Cette variable est une variable Flow et retourne une chaîne de caractères si la colonne est configurée à Key. Sinon, elle ne retourne rien.</td>
</tr>
<tr>
<td>NB_COLUMN</td>
<td>Nombre de colonnes itérées jusqu'à présent. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Il faut utiliser ce composant en association avec les autres composants MS SQL, notamment avec le tMSSqlConnection.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé, consultez Scénario : Itérer une table de base de données et lister le nom des colonnes de la table à la page 2611.
tMSSqlCommit

Ce composant utilise une connexion unique pour commiter en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

Le tMSSqlCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tMSSqlCommit Standard

Ces propriétés sont utilisées pour configurer le tMSSqlCommit s’exécutant dans le framework de Jobs Standard.

Le composant tMSSqlCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Base setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tMSSqlConnection dans la liste.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. <strong>Avertissement :</strong> Si vous utilisez un lien de type Row &gt; Main pour relier le tMSSqlCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Advanced setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>
**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants MSSql et notamment avec les composants tMSSqlConnection et tMSSqlRollback.</th>
</tr>
</thead>
</table>

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend. |

**Scénario associé**

Pour un scénario associé au composant tMSSqlCommit, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tMSSqlConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Le tMSSqlConnection ouvre une connexion vers une base de données Microsoft SQL Server ou une base de données Microsoft Azure SQL.

**Propriétés du tMSSqlConnection Standard**

Ces propriétés sont utilisées pour configurer le tMSSqlConnection s’exécutant dans le framework de Jobs Standard.

Le composant tMSSqlConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>JDBC Provider</td>
<td>Sélectionnez le fournisseur du pilote JDBC à utiliser, Microsoft (recommandé) ou Open source JTDS.</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Lorsque vous utilisez ce composant avec la source de données dans Talend Runtime, vous devez utiliser le pilote Open source JTDS.</td>
</tr>
<tr>
<td>Open source JTDS</td>
<td>Notez que, quand Microsoft est sélectionné, vous devez télécharger le pilote Microsoft JDBC pour les serveurs SQL sur Microsoft Download Center (en anglais). Vous devez ensuite décompresser le fichier zip téléchargé, choisir un fichier .jar dans le dossier décompressé, selon votre version de JRE, renommer le fichier .jar en mssql-jdbc.jar et l’installer manuellement. Pour plus d’informations relatives au choix du fichier .jar, consultez la configuration système requise sur Microsoft Download Center (en anglais).</td>
</tr>
</tbody>
</table>
### Adresse IP du serveur de base de données.

**Host**

Numéro du port d’écoute du serveur de base de données.

**Port**

Nom du schéma.

**Schema**

Nom de la base de données.

**Database**

Informations d’authentification de l’utilisateur de base de données.

**Username et Password**

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Additional JDBC parameters**

Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, `encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;` pour une connexion à la base de données Azure SQL.

**Use or register a shared DB Connection**

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un composant **tRunJob** ayant une de ces options activée fera échouer votre Job.

Cette case est indisponible lorsque la case **Specify a data source alias** est cochée.

**Specify a data source alias**

Cochez cette case et spécifiez l’alias de la source de données créée dans **Talend Runtime** pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans **Talend Runtime**. Pour voir un exemple de cas d’utilisation, consultez **Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL** à la page 2647.

Cette option est indisponible lorsque la case **Use an existing connection** est cochée.

**Data source alias**

Saisissez l’alias de la source de données créée du côté de **Talend Runtime**.
Advanced settings

**Auto Commit**

Cochez cette case afin de committer automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de committer chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne committe pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé avec des composants MSSql, notamment les composants `tMSSqlCommit` et `tMSSqlRollback`.

**Limitation**


Scénario : Insérer des données dans une table de base de données et en extraire des informations

Ce scénario décrit un Job lisant des données d’un fichier texte, relatives aux employés, insérant les données dans une table d’une base de données MSSQL et extrayant les données les plus utiles de la table, pour les afficher dans la console.
Ce scénario comprend les composants suivants :

- un **tMSSqlConnection** : établit une connexion au serveur MSSQL.
- un **tFileInputDelimited** : lit le fichier d’entrée, définit la structure des données et les envoie au composant suivant.
- un **tMSSqlOutput** : écrit les données qu’il reçoit du composant précédent dans une table d’une base de données MSSQL.
- un **tMSSqlInput** : extrait les données de la table selon une requête SQL.
- un **tLogRow** : affiche les informations reçues du composant précédent dans la console.
- un **tMSSqlCommit** : commite la transaction sur le serveur MSSQL connecté.

### Construire le Job

### Procédure

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un **tMSSqlConnection**, un **tFileInputDelimited**, un **tMSSqlOutput**, un **tMSSqlInput**, un **tLogRow** et un **tMSSqlCommit**.
2. Connectez le **tMSSqlConnection** au **tFileInputDelimited** à l’aide d’un lien **Trigger > OnSubjobOk**.
3. Reliez de la même manière le **tFileInputDelimited** au **tMSSqlInput** et le composant **tMSSqlInput** au **tMSSqlCommit**.
4. Connectez le **tFileInputDelimited** au **tMSSqlOutput** à l’aide d’un lien **Row > Main**.
5. Répétez l’opération pour connecter le **tMSSqlInput** au **tLogRow**.
**Configurer les composants**

**Ouvrir une connexion au serveur MSSQL**

**Procédure**

1. Double-cliquez sur le composant `tMSSqlConnection` pour ouvrir sa vue `Basic settings`.

   ![tMSSqlConnection](image)

   2. Dans le champ `Host`, saisissez l'adresse IP ou le nom de l'hôte du serveur MSSQL, 192.168.30.47 dans cet exemple.

   3. Dans le champ `Port`, saisissez le numéro du port du serveur de la base de données, 1433 dans ce scénario.

   4. Dans le champ `Schema`, saisissez le nom du schéma, `dbo` dans cet exemple.

   5. Dans le champ `Database`, saisissez le nom de la base de données, `talend` dans ce scénario.

   6. Dans les champs `Username` et `Password`, saisissez respectivement votre identifiant et votre mot de passe de connexion à MSSQL.

**Lire les données d'entrée**

**Procédure**

1. Double-cliquez sur le composant `tFileInputDelimited` pour ouvrir sa vue `Component`.

   ![tFileInputDelimited](image)

   2. Cliquez sur le bouton ` [...] ` à côté du champ `File Name/Stream` afin de parcourir votre système jusqu'au fichier d'entrée, par exemple `D:/Input/Employee_Wage.txt`. Ce fichier texte contient trois co
lonnes : *id, name et wage*. id;name;wage 51;Harry;2300 40;Ronald;3796 17;Theodore;2174 21;James;1986 2;George;2591 89;Calvin;2362 84;Ulysses;3383 4;Lyndon;2264 17;Franklin;1780 86;Lyndon;3999

3. Dans le champ **Header**, saisissez 1 pour ignorer la première ligne du fichier d’entrée.

4. Cliquez sur **Edit schema** pour définir les données à passer au composant **tMSSqlOutput**. Dans cet exemple, définissez *id* comme la clé (cochez la case **Key**) et spécifiez la longueur (**Length**) et la précision (**Precision**) pour chaque colonne.

Cliquez sur **OK** pour fermer l’éditeur de schéma. Une boîte de dialogue s’ouvre, dans laquelle vous pouvez choisir de propager le schéma au composant suivant.

Pour plus d’informations, consultez **tFileInputDelimited** à la page 1067.

**Écrire les données dans la table de la base de données**

**Procédure**

1. Double-cliquez sur le **tMSSqloOutput** pour ouvrir sa vue **Basic settings**.

2. Saisissez les informations requises pour la connexion ou utilisez une connexion existante précédemment configurée. Dans cet exemple, cochez la case **Use an existing connection**. Si plusieurs connexions sont disponibles, sélectionnez celle que vous souhaitez utiliser, dans la liste déroulante **Component List**.

3. Dans le champ **Table**, saisissez le nom de la table dans laquelle vous souhaitez écrire les données : *Wage_Info* dans ce scénario. Vous pouvez également cliquer sur le bouton [...] à côté du champ **Table** pour ouvrir une boîte de dialogue et sélectionner une autre table.

4. Sélectionnez **Create table if not exists** dans la liste déroulante **Action on table**.
5. Sélectionnez Insert if not exists dans la liste déroulante Action on data.
6. Cliquez sur Sync columns pour récupérer le schéma du composant précédent.

**Extrait des informations de la table**

**Procédure**

1. Double-cliquez sur le composant tMSSqlInput pour afficher sa vue Component.

   ![Image de tMSSqlInput](image)

   **Fenêtre de Configuration**
   - **Use an existing connection** cochée
   - **Table Name**: "Wage_Info"
   - **Query**:
     ```sql
     SELECT * 
     FROM Wage_Info 
     WHERE wage > (SELECT avg(wage) 
                  FROM Wage_Info) 
     ORDER BY id
     ```

2. Cochez la case Use an existing connection. Si plusieurs connexions sont disponibles, sélectionnez celle que vous souhaitez utiliser, dans la liste déroulante Component List.
3. Cliquez sur Edit schema pour définir la structure des données à lire depuis la table. Dans cet exemple, vous devez lire les trois colonnes de la table.

   ![Image de tMSSqlInput Schema](image)

4. Dans le champ Table Name, saisissez le nom de la table de laquelle vous souhaitez lire les données : Wage_Info dans ce scénario.
5. Dans le champ **Query**, renseignez la requête SQL à exécuter sur la table spécifiée. Pour obtenir les données relatives aux employés dont le salaire est supérieur à la moyenne et les classer par ID, saisissez la requête SQL comme suit:

```
SELECT * FROM Wage_Info WHERE wage > (SELECT avg(wage) FROM Wage_Info) ORDER BY id
```

**Afficher les informations dans la console**

**Procédure**

1. Double-cliquez sur le **tLogRow** pour afficher sa vue **Basic settings**.
2. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)**.

**Committer la transaction et fermer la connexion**

**Procédure**

1. Double-cliquez sur le **tMSSqlCommit** pour ouvrir sa vue **Basic settings**.
2. Cochez la case **Close Connection**.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.
2. Exécutez le Job en appuyant sur la touche **F6** ou en cliquant sur le bouton **Run** de la vue **Run**.

La console affiche l'ID des employés dont le salaire est supérieur à la moyenne, leur nom et leur salaire. Les employés sont classés par ID.

```
<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>wage</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>Ronald</td>
<td>3796</td>
</tr>
<tr>
<td>84</td>
<td>Ulysses</td>
<td>3383</td>
</tr>
<tr>
<td>96</td>
<td>Lyndon</td>
<td>3999</td>
</tr>
</tbody>
</table>
```


**tMSSqlInput**

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma.

Le tMSSqlInput lit des données et en extraît des champs à l’aide de requêtes d’une base de données Microsoft SQL Server ou d’une base de données Microsoft Azure SQL. Les champs récupérés sont ensuite transmis au composant suivant via une connexion de flux (Main row).

**Propriétés du tMSSqlInput Standard**

Ces propriétés sont utilisées pour configurer le tMSSqlInput s’exécutant dans le framework de Jobs Standard.

Le composant tMSSqlInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.</strong></td>
<td></td>
</tr>
</tbody>
</table>
Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.

| **JDBC Provider** | Sélectionnez le fournisseur du pilote JDBC à utiliser, **Microsoft** (recommandé) ou **Open source JTDS**.  
Lorsque vous utilisez ce composant avec la source de données dans Talend Runtime, vous devez utiliser le pilote **Open source JTDS**.  
Notez que, quand **Microsoft** est sélectionné, vous devez télécharger le pilote Microsoft JDBC pour les serveurs SQL sur Microsoft Download Center (en anglais). Vous devez ensuite décompresser le fichier zip téléchargé, choisir un fichier .jar dans le dossier décompressé, selon votre version de JRE, renommer le fichier .jar en msssql-jdbc.jar et l’installer manuellement. Pour plus d’informations relatives au choix du fichier .jar, consultez la configuration système requise sur Microsoft Download Center (en anglais). |
| **Host** | Adresse IP du serveur de base de données. |
| **Port** | Numéro du port d’écoute du serveur de base de données. |
| **Database** | Nom de la base de données. |
| **Schema** | Nom du schéma. |
| **Username et Password** | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma. |
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th>Table name</th>
<th>Nom de la table à lire.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query type et Query</td>
<td>Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs correspond à celui défini dans le schéma.</td>
</tr>
<tr>
<td>Specify a data source alias</td>
<td>Cochez cette case et spécifiez l’alias de la source de données créée dans Talend Runtime pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647. Cette option est indisponible lorsque la case <strong>Use an existing connection</strong> est cochée.</td>
</tr>
<tr>
<td>Data source alias</td>
<td>Saisissez l’alias de la source de données créée du côté de Talend Runtime. Ce champ est disponible uniquement lorsque la case <strong>Specify a data source alias</strong> est cochée.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| Additional JDBC parameters | Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, `encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;` pour une connexion à la base de données Azure SQL. Ce champ n’est pas disponible si la case **Use an existing connection** est cochée. |
| Trim all the String/Char columns | Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères. |
| Trim column | Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées. |
### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données MS SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.</td>
</tr>
<tr>
<td></td>
<td>Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de</td>
</tr>
</tbody>
</table>

### Scénarios associés

Pour des scénario associés, consultez :

- tMSSqlConnection à la page 2545.

Consultez également le scénario du tContextLoad Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520.
tMSSqlLastInsertld

Ce composant récupère les dernières clés primaires ajoutées par un utilisateur à une table MSSql.
Le tMSSqlLastInsertld affiche les derniers ID ajoutés à une table à partir d’une connexion MSSql spécifiée.

Propriétés du tMSSqlLastInsertld Standard

Ces propriétés sont utilisées pour configurer le tMSSqlLastInsertld s’exécutant dans le framework de Jobs Standard.
Le composant tMSSqlLastInsertld Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les</td>
</tr>
</tbody>
</table>
modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

| Component list | Sélectionnez le composant tMSSqlConnection dans la liste s’il y a plus d’une connexion dans votre Job. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. |
Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé au composant **tMSSqlLastInsertId**, consultez Scénario : Récupérer les ID des dernières entrées ajoutées avec le **tMysqlLastInsertId** à la page 2655.
Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.
Le tMSSqlOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

**Propriétés du tMSSqlOutput Standard**

Ces propriétés sont utilisées pour configurer le tMSSqlOutput s’exécutant dans le framework de Jobs Standard.
Le composant tMSSqlOutput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

**Remarque** :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
</tbody>
</table>
| **Configurer une connexion existante** | Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant.
Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le **Guide utilisateur du Studio Talend**. |
| **Use an existing connection** | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |
| **Remarque** | **Remarque** :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existante entre les deux niveaux, par |
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

### JDBC Provider

Sélectionnez le fournisseur du pilote JDBC à utiliser, **Microsoft** (recommandé) ou **Open source JTDS**.

Lorsque vous utilisez ce composant avec la source de données dans Talend Runtime, vous devez utiliser le pilote **Open source JTDS**.

Notez que, quand **Microsoft** est sélectionné, vous devez télécharger le pilote Microsoft JDBC pour les serveurs SQL sur Microsoft Download Center (en anglais). Vous devez ensuite décompresser le fichier zip téléchargé, choisir un fichier .jar dans le dossier décompressé, selon votre version de JRE, renommer le fichier .jar en `mssql-jdbc.jar` et l’installer manuellement. Pour plus d’informations relatives au choix du fichier .jar, consultez la configuration système requise sur Microsoft Download Center (en anglais).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
</tbody>
</table>
| **Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :

  - **Default** : n’effectuer aucune opération de table.
  - **Drop and create table** : supprimer la table puis en créer une nouvelle.
  - **Create table** : créer une table qui n’existe pas encore. |
<table>
<thead>
<tr>
<th>Create table if not exists</th>
<th>créer la table si nécessaire.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop table if exists and create</td>
<td>supprimer la table si elle existe déjà, puis en créer une nouvelle.</td>
</tr>
<tr>
<td>Clear table</td>
<td>supprimer le contenu de la table.</td>
</tr>
<tr>
<td>Truncate table</td>
<td>supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.</td>
</tr>
</tbody>
</table>

**Turn on identity insert**

Cochez cette case pour utiliser votre propre séquence sur les valeurs Identity des données insérées (plutôt que de laisser le serveur SQL choisir les valeurs séquentielles).

**Action on data**

Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

- **Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
- **Single Insert Query** : Ajouter de nouvelles entrées à la table, regroupées dans un lot.
- **Update** : Mettre à jour les entrées existantes.
- **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
- **Delete** : Supprimer les entrées correspondantes au flux d’entrée.
- **Insert if not exist** : Ajouter de nouvelles entrées à la table si nécessaire.

**Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les cases correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

**Specify identity field**

Cochez cette case pour spécifier quel est le champ identity (**Identity field**), constitué d’un numéro d’identification incrémenté automatiquement. Lorsque cette case est cochée, trois autres champs s’affichent :
### Identity field

Sélectionnez dans la liste la colonne que vous souhaitez définir comme champ identity.

### Start value

Saisissez une valeur de départ, utilisée pour la première ligne chargée dans la table.

### Step

Saisissez une valeur d’incrémentation ajoutée à la valeur de la ligne précédemment chargée.

Cette case est disponible si vous sélectionnez l’option **Drop and create table**, **Create table**, **Create table if not exists** ou **Drop table if exists and create** dans la liste **Action on table**. Elle ne s’affiche pas si vous sélectionnez l’option **Enable parallel execution** dans la vue **Advanced settings**. Si vous cochez cette case sans avoir cochée la case **Turn on identity insert**, mais en ayant sélectionné **Create table if not exists** dans la liste **Action on table** et si la table spécifiée n’existe pas, seule une table est créée sans qu’aucune donnée y soit insérée.

**Remarque :**

Vous pouvez également spécifier le champ identity (**identity field**) à partir du schéma du composant. Pour ce faire, paramétrez le type de base de données (**DB Type**) de la colonne correspondante en **INT IDENTITY**.

**Remarque :**

Lorsque la case **Specify identity field** est cochée, le type de base de données (**DB Type**) **INT IDENTITY** du schéma est ignoré.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.
**Repository**: Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

---

### Specify a data source alias

Cochez cette case et spécifiez l’alias de la source de données créée dans Talend Runtime pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647.

Cette option est indisponible lorsque la case Use an existing connection est cochée.

### Data source alias

Saisissez l’alias de la source de données créée du côté de Talend Runtime.

Ce champ est disponible uniquement lorsque la case Specify a data source alias est cochée.

### Die on error

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects.

---

### Advanced settings

#### Additional JDBC parameters

Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, `encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;` pour une connexion à la base de données Azure SQL.

Ce champ n’est pas disponible si la case Use an existing connection est cochée.

#### Commit every

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

#### Additional Columns

Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement
Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.

**Name** : Saisissez le nom de la colonne à modifier ou à insérer.

**SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.

**Position** : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l’action à effectuer sur la colonne de référence.

**Reference column** : Saisissez une colonne de référence que le composant **tMSSqlOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

**Use field options** : Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.

**Ignore date validation** : Cochez cette case pour ignorer la validation de la date et insérer les données directement dans la base de données pour les types de données DATE, DATETIME, DATETIME2 et DATETIMEOFFSET.

**Enable debug mode** : Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.

**Support null in “SQL WHERE” statement** : Cochez cette case pour prendre en compte les valeurs Null d’une table de base de données.

**tStatCatcher Statistics** : Cochez cette case pour collecter les données de log au niveau du composant.

**Use Batch** : Cochez cette case pour activer le mode de traitement par lots pour le traitement des données. Cette case est disponible uniquement si vous avez choisi l’option **Insert**, **Update**, **Single Insert Query** ou **Delete** dans la liste **Action on data**.

**Batch Size** : Spécifiez le nombre d’enregistrements à traiter dans chaque lot.
Ce champ est disponible uniquement lorsque la case **Use batch mode** est cochée.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>NB_LINE</strong> updated : nombre de lignes mises à jour. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_LINE</strong> inserted : nombre de lignes insérées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_LINE</strong> deleted : nombre de lignes supprimées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_LINE</strong> rejected : nombre de lignes rejetées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td><strong>QUERY</strong> : requête traitée. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données MSSQL. Il permet aussi de créer un flux de rejet avec un lien <strong>Row &gt; Reject</strong> filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant <strong>tMysqlOutput</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton <strong>[+]</strong> pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant</td>
</tr>
</tbody>
</table>
### Scénarios associés

Pour un scénario associé au composant `tMSSqlOutput`, consultez :

- `tMSSqlConnection` à la page 2545.

- Scénario : Insérer une colonne et modifier les données en utilisant le `tMSSqlOutput` à la page 2667.

<table>
<thead>
<tr>
<th>Limitation</th>
</tr>
</thead>
</table>
tMSSqlOutputBulk

Ce composant prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données MSSQL.

Les composants tMSSqlOutputBulk et tMSSqlBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tMSSqlOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Le tMSSqlOutputBulk écrit un fichier composé de colonnes et basé sur le séparateur défini et sur les standards MSSql.

Propriétés du tMSSqlOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tMSSqlOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tMSSqlOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom du fichier à générer.</td>
</tr>
<tr>
<td></td>
<td><strong>Avertissement</strong> :</td>
</tr>
<tr>
<td></td>
<td>Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.</td>
</tr>
<tr>
<td>Append</td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
</tbody>
</table>
### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Advanced settings

<table>
<thead>
<tr>
<th>Row separator</th>
<th>Chaîne (ex : \n sous Unix) séparant les lignes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field separator</td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td>Include header</td>
<td>Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez <strong>Custom</strong> et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</td>
</tr>
</tbody>
</table>
tMSSqlOutputBulk

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec le composant tMSSQLBulkExec. Ensemble, ils offrent un gain de performance important pour l’alimentation d’une base de données MSSQL. |

**Scénarios associés**

Pour un scénario associé au tMSSqlOutputBulk, consultez :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du composant tMysqlOutputBulk.
- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec.
tMSSqlOutputBulkExec

Ce composant améliore les performances pendant les opérations d’Insert dans une base de données Microsoft SQL Server.

Les composants tMSSqlOutputBulk et tMSSqlBulkExec sont généralement utilisés ensemble comme deux parties d’un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l’opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tMSSqlOutputBulkExec.

Propriétés du tMSSqlOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tMSSqlOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tMSSqlOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Permet de choisir entre les options Bulk Insert et Bulk update.</td>
</tr>
<tr>
<td>Property Type</td>
<td>Le schéma peut être Built-in ou distant dans le Repository.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.</td>
</tr>
<tr>
<td>JDBC Provider</td>
<td>Sélectionnez le fournisseur du pilote JDBC à utiliser, Microsoft (recommandé) ou Open source JTDS.</td>
</tr>
</tbody>
</table>
Lorsque vous utilisez ce composant avec la source de données dans Talend Runtime, vous devez utiliser le pilote **Open source JTDS**.

Notez que, quand **Microsoft** est sélectionné, vous devez télécharger le pilote Microsoft JDBC pour les serveurs SQL sur [Microsoft Download Center](en anglais). Vous devez ensuite décompresser le fichier zip téléchargé, choisir un fichier .jar dans le dossier décompressé, selon votre version de JRE, renommer le fichier .jar en `mssql-jdbc.jar` et l’installer manuellement. Pour plus d’informations relatives au choix du fichier .jar, consultez la configuration système requise sur [Microsoft Download Center](en anglais).

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de la base de données. Seules l’adresse IP exacte de la machine locale, localhost ou 127.0.0.1 permettent un fonctionnement optimal. Le serveur de la base de données doit être installé sur la même machine que le Studio Talend ou que le Job contenant un <code>tMSSqlOutputBulkExec</code>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit déjà exister pour que l’opération d’insert soit autorisée.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
- **None** : n’effectuer aucune opération de table.  
- **Drop and create the table** : supprimer la table puis en créer une nouvelle.  
- **Create a table** : créer une table qui n’existe pas encore.  
- **Create table if doesn’t exist** : créer la table si nécessaire.  
- **Clear a table** : supprimer le contenu de la table.  
- **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark,
évitez le mot réservé `line` lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

**Repository** : Le schéma existe déjà et il est stocké dans le *Repository*. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Local File Name**
Nom du fichier à générer et à charger.
Le fichier est généré sur la machine spécifiée par l’URI dans le champ *Host* et doit être sur la même machine que le serveur de la base de données.

**Append**
Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.

**Advanced settings**

**Additional JDBC parameters**
Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, `encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;` pour une connexion à la base de données Azure SQL.
<table>
<thead>
<tr>
<th>Champs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field separator</td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td>Row separator</td>
<td>Chaîne (ex: <code>\n</code> sous Unix) séparant les lignes.</td>
</tr>
<tr>
<td>First row</td>
<td>Saisissez le numéro du rang à partir duquel vous voulez démarrer l’action.</td>
</tr>
<tr>
<td>Include header</td>
<td>Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.</td>
</tr>
<tr>
<td>Data file type</td>
<td>Sélectionnez le type de données à traiter.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans la base de données.</th>
</tr>
</thead>
</table>
| Dynamic settings   | Cliquez sur le bouton `[+]` pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les
<table>
<thead>
<tr>
<th>Limitation</th>
<th>Le serveur de la base de données doit être installé sur la même machine que le <em>Studio Talend</em> ou le Job contenant un <em>tMSSqlOutputBulkExec</em>, afin que le composant fonctionne correctement.</th>
</tr>
</thead>
</table>

### Scénarios associés

Pour un scénario associé au *tMSSqlOutputBulkExec*, consultez :

- *Scénario : Insérer des données transformées dans une base MySQL* à la page 2685 du composant *tMysqlOutputBulk*.

- *Scénario : Insérer des données en masse dans une base MySQL* à la page 2692 du *tMysqlOutputBulkExec*. 

paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*. 

**tMSSqlRollback**

Ce composant annule le commit de la transaction dans la base de données MS SQL connectée.

**Propriétés du tMSSqlRollback Standard**

Ces propriétés sont utilisées pour configurer le tMSSqlRollback s’exécutant dans le framework de Jobs Standard.

Le composant tMSSqlRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tMSSqlConnection dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec d’autres composants MSSql, notamment les composants tMSSqlConnection et tMSSqlCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez...</td>
</tr>
</tbody>
</table>
pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Pour un scénario associé, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623.
tMSSqlRow

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler).

Le tMSSqlRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée Microsoft SQL Server ou Azure SQL, selon la nature de la requête et de la base de données. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Propriétés du tMSSqlRow Standard

Ces propriétés sont utilisées pour configurer le tMSSqlRow s’exécutant dans le framework de Jobs Standard.

Le composant tMSSqlRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th></th>
<th>Séléctionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>JDBC Provider</td>
<td>Sélectionnez le fournisseur du pilote JDBC à utiliser, Microsoft (recommandé) ou Open source JTDS.</td>
</tr>
</tbody>
</table>
Lorsque vous utilisez ce composant avec la source de données dans Talend Runtime, vous devez utiliser le pilote **Open source JTDS**.

Notez que, quand **Microsoft** est sélectionné, vous devez télécharger le pilote Microsoft JDBC pour les serveurs SQL sur **Microsoft Download Center** (en anglais). Vous devez ensuite décompresser le fichier zip téléchargé, choisir un fichier .jar dans le dossier décompressé, selon votre version de JRE, renommer le fichier .jar en `mssql-jdbc.jar` et l’installer manuellement. Pour plus d’informations relatives au choix du fichier .jar, consultez la configuration système requise sur **Microsoft Download Center** (en anglais).

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <code>line</code> lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
|                  | Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
|                  | • **View schema** : sélectionnez cette option afin de voir le schéma. |
|                  | • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. |
|                  | • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez |
propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métdonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois et la table doit exister pour que l’opération d’Insert soit autorisée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn on identity insert</td>
<td>Cochez cette case pour utiliser votre propre séquence sur les valeurs Identity des données insérées (plutôt que de laisser le serveur SQL choisir les valeurs séquentielles).</td>
</tr>
<tr>
<td>Query type</td>
<td>La requête peut être <strong>Built-in</strong> ou distante dans le Repository.</td>
</tr>
<tr>
<td>Query</td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td>Specify a data source alias</td>
<td>Cochez cette case et spécifiez l’alias de la source de données créée dans <strong>Talend Runtime</strong> pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans <strong>Talend Runtime</strong>. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans <strong>Talend Runtime</strong> pour récupérer les données d'une base de données MySQL à la page 2647. Cette option est indisponible lorsque la case <strong>Use an existing connection</strong> est cochée.</td>
</tr>
<tr>
<td>Data source alias</td>
<td>Saisissez l’alias de la source de données créée du côté de <strong>Talend Runtime</strong>. Ce champ est disponible uniquement lorsque la case <strong>Specify a data source alias</strong> est cochée.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien <strong>Row &gt; Rejects</strong>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| Additional JDBC parameters | Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et |
chaque propriété est une paire clé-valeur. Par exemple, encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30; pour une connexion à la base de données Azure SQL.
Ce champ n’est pas disponible si la case Use an existing connection est cochée.

**Propagate QUERY’s recordset**

Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column.

`${i}`

**Remarque :**

Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type `Object`. Ce composant est généralement suivi du tParseRecordSet.

**Use PreparedStatement**


- **Parameter Index** : Saisissez la position du paramètre dans l’instruction SQL.
- **Parameter Type** : Saisissez le type du paramètre.
- **Parameter Value** : Saisissez la valeur du paramètre.

`${i}`

**Remarque :**

Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

**Commit every**

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>QUERY</strong> : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
</table>
| **Dynamic settings**      | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.  

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez  **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**. |
| **Limitation**            | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com). |
Scénarios associés

Pour un scénario associé, consultez :

- Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
- Procédure du composant tDBSQLRow.
- Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du tMysqlRow.
tMSSqlSCD

Ce composant reflète et traque les modifications d'une table SCD dédiée sur un serveur Microsoft SQL ou dans une base de données Azure SQL.

Le tMSSqlSCD répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.

Propriétés du tMSSqlSCD Standard

Ces propriétés sont utilisées pour configurer le tMSSqlSCD s'exécutant dans le framework de Jobs Standard.

Le composant tMSSqlSCD Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

| JDBC Provider | Sélectionnez le fournisseur du pilote JDBC à utiliser, Microsoft (recommandé) ou Open source JTDS.
|               | Lorsque vous utilisez ce composant avec la source de données dans Talend Runtime, vous devez utiliser le pilote Open source JTDS.
|               | Notez que, quand Microsoft est sélectionné, vous devez télécharger le pilote Microsoft JDBC pour les serveurs SQL sur Microsoft Download Center (en anglais). Vous devez ensuite décompresser le fichier zip téléchargé, choisir un fichier .jar dans le dossier décompressé, selon votre version de JRE, renommer le fichier .jar en mssql-jdbc.jar et l'installer manuellement. Pour plus d'informations relatives au choix du fichier .jar, consultez la configuration système requise sur Microsoft Download Center (en anglais). |
| Server        | Adresse IP du serveur de base de données. |
| Port          | Numéro du port d'écoute du serveur de base de données. |
| Schema        | Nom du schéma de la base de données. |
| Database      | Nom de la base de données. |
| Username et Password | Informations d’authentification de l’utilisateur de la base de données. 
|               | Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Table         | Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. 
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles : |

- **View schema** : sélectionnez cette option afin de voir le schéma.
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


**SCD Editor**

L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension.

Pour plus d’informations concernant les méthodes de création, consultez Méthodologie de gestion du SCD à la page 2716.

**Use memory saving Mode**

Cochez cette case pour améliorer les performances du système.

**Source keys include Null**

Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null.

**Avertissement** :

Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.

**Die on error**

Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

**Advanced settings**

**Additional JDBC parameters**

Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, `encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;` pour une connexion à la base de données Azure SQL.

Ce champ n’est pas disponible si la case **Use an existing connection** est cochée.

**End date time details**

Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format **HH:mm:ss**. La valeur par défaut pour ce champ est **12:00:00**.
Ce champ apparaît uniquement lorsqu’un SCD de Type 2 est utilisé et lorsque Fixed year value est sélectionné pour créer la date de fin du SCD.

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug mode</td>
<td>Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.</td>
</tr>
</tbody>
</table>

### Variables globales

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d´un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

| Limitation | Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD. |

**Scénario associé**

Pour un scénario associé, consultez **tMysqlSCD** à la page 2712.
**tMSSqlSP**

Ce composant permet de centraliser des requêtes multiples ou complexes dans une base de données et de les appeler plus facilement.

Le tMSSqlSP appelle une procédure stockée dans une base de données Microsoft SQL Server ou Azure SQL.

**Propriétés du tMSSqlSP Standard**

Ces propriétés sont utilisées pour configurer le tMSSqlSP s’exécutant dans le framework de Jobs Standard.

Le composant tMSSqlSP Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier des propriétés du composant. Les champs suivants sont alors pré-remplis.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>JDBC Provider</strong></td>
<td>Sélectionnez le fournisseur du pilote JDBC à utiliser, <strong>Microsoft</strong> (recommandé) ou <strong>Open source JTDS</strong>.</td>
</tr>
<tr>
<td></td>
<td>Lorsque vous utilisez ce composant avec la source de données dans Talend Runtime, vous devez utiliser le pilote <strong>Open source JTDS</strong>.</td>
</tr>
<tr>
<td></td>
<td>Notez que, quand <strong>Microsoft</strong> est sélectionné, vous devez télécharger le pilote Microsoft JDBC pour les serveurs SQL sur <strong>Microsoft Download Center</strong> (en anglais). Vous devez ensuite décompresser le fichier zip téléchargé.</td>
</tr>
</tbody>
</table>
choisir un fichier .jar dans le dossier décompressé, selon votre version de JRE, renommer le fichier .jar en mssql-jdbc.jar et l'installer manuellement. Pour plus d'informations relatives au choix du fichier .jar, consultez la configuration système requise sur Microsoft Download Center (en anglais).

| Host | Adresse IP du serveur de base de données. |
| Port | Numéro du port d'écoute du serveur de base de données. |
| Database | Nom de la base de données. |
| Schema | Nom du schéma. |
| Username et Password | Informations d'authentification sur l'utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. 

**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

**Repository** : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il est réutilisable. Voir également le Guide utilisateur du Studio Talend.  

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**SP Name**

Saisissez le nom exact de la Procédure Stockée (SP).
| **Is Function / Return result in** | Cochez cette case, si une seule valeur doit être retournée. Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir. |
| **Parameters** | Cliquez sur le bouton [+], et sélectionnez dans le champ Schema Columns les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu’il n’y a de paramètres utilisés dans la procédure. Sélectionnez le Type de paramètre : IN : paramètre d’entrée (Input) OUT : paramètre de sortie (Output)/valeur retournée IN OUT : les paramètres d’entrée doivent être retournées sous forme de valeur, même après modifications via la procédure (fonction). RECORDSET : les paramètres d’entrée doivent être retournées sous forme d’ensemble de valeurs, au lieu d’une valeur unique. Remarque : Consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620 si vous voulez analyser un ensemble d’enregistrements d’une table de données ou d’une requête SQL. |
| **Specify a data source alias** | Cochez cette case et spécifiez l’alias de la source de données créée dans Talend Runtime pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647. Cette option est indisponible lorsque la case Use an existing connection est cochée. |
| **Data source alias** | Saisissez l’alias de la source de données créée du côté de Talend Runtime. Ce champ est disponible uniquement lorsque la case Specify a data source alias est cochée. |

**Advanced settings**

<p>| <strong>Additional JDBC parameters</strong> | Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30; pour une connexion à la base de données Azure SQL. |</p>
<table>
<thead>
<tr>
<th><strong>tMSSqlSP</strong></th>
<th>Ce champ n'est pas disponible si la case <em>Use an existing connection</em> est cochée.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th><strong>Règle d’utilisation</strong></th>
<th>Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d’entrée sont autorisés.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <em>Code</em>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <em>Dynamic settings</em> est disponible uniquement lorsque la case <em>Use an existing connection</em> est cochée dans la vue <em>Basic settings</em>. Lorsqu’un paramètre dynamique est configuré, la liste <em>Component List</em> de la vue <em>Basic settings</em> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

|----------------|---------------------------------------------------------------------------------------------------------------------------------|
Scénario : Récupérer des informations personnelles à l’aide d’une procédure stockée

Ce scénario décrit un Job récupérant un enregistrement contenant des informations personnelles à partir d’une table spécifiée. Il utilise pour cela une procédure stockée se basant sur la valeur id définie dans le flux d’entrée.

Dans ce scénario, la table de laquelle vous souhaitez récupérer des données contient des informations personnelles. Pour reproduire ce scénario, vous pouvez écrire les données dans une table à partir d’un fichier CSV, comme suit. Pour plus d’informations concernant l’écriture de données dans une table MS SQL, consultez Scénario : Insérer des données dans une table de base de données et en extraire des informations à la page 2547.

Dans ce scénario, la procédure stockée utilisée pour récupérer les informations personnelles se présente comme suit :

```
CREATE PROCEDURE [dbo].[QueryPerson]
@id int,
@name varchar(50)
AS
BEGIN
SET NOCOUNT ON
SELECT * FROM dbo.person where id=@id
END
```

Construire le Job

Procédure

1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant à partir de la Palette : un tFixedFlowInput, un tMSSqIsp, un tParseRecordSet et deux composants tLogRow.
2. Reliez le tFixedFlowInput au tMSSqIsp à l’aide d’un lien Row > Main.
3. Répétez l’opération pour connecter le tMSSqIsp au premier tLogRow, le premier tLogRow au tParseRecordSet et le tParseRecordSet au second tLogRow.
Configurer les composants

Configurer le composant d’entrée

Procédure

1. Double-cliquez sur le composant tFixedFlowInput pour ouvrir sa vue Basic settings.

![Schema de tFixedFlowInput](image)

2. Cliquez sur le bouton [...] à côté du champ Edit schema pour ouvrir l’éditeur du schéma.

![Éditeur du schéma de tFixedFlowInput](image)

Cliquez deux fois sur le bouton [+ ] pour ajouter deux colonnes : id de type Integer et name de type String.
Cliquez sur OK pour fermer l’éditeur du schéma.

3. Dans la zone Mode, renseignez chaque colonne avec sa valeur respective. Dans cet exemple, la valeur de la colonne id est 4 et celle de la colonne name est null car elle n’est pas utilisée dans l’instruction SQL SELECT dans la procédure stockée.

Configurer le composant tMSSqlSP

Procédure

1. Double-cliquez sur le tMSSqlSP pour ouvrir sa vue Basic settings.
2. Dans les champs **Host**, **Port**, **Schema**, **Database**, **Username** et **Password**, renseignez vos informations de connexion au serveur MS SQL.

3. Dans le champ **SP Name**, saisissez le nom de la procédure stockée à utiliser. Par exemple, saisissez **QueryPerson**.

4. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur du schéma.

Cliquez sur le bouton pour copier toutes les colonnes du schéma d’entrée dans le schéma de sortie. Cliquez sur le bouton [] du panneau de droite afin d’ajouter une nouvelle colonne **person** de type **Object** qui contiendra les informations personnelles à récupérer de la base de données. Cliquez sur **OK** pour fermer l’éditeur du schéma et acceptez la propagation proposée par la boîte de dialogue.
5. Dans la zone **Parameters**, cliquez trois fois sur le bouton [+] pour ajouter trois lignes et sélectionnez une colonne de schéma ainsi que son type pour chaque ligne. Dans cet exemple, les colonnes *id* et *name* sont de type **IN**, et la colonne *person* est de type **RECORD SET**.

**Configurer le tParseRecordSet**

**Procédure**

1. Double-cliquez sur le **tParseRecordSet** pour ouvrir sa vue **Basic settings**.

   ![tParseRecordSet](Image)


   3. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur du schéma.

      ![Schema of tParseRecordSet](Image)

      Cliquez quatre fois sur le bouton [+] dans le panneau de droite pour ajouter quatre colonnes : *id* et *age* de type **Integer**, et *name* et *sex* de type **String**.

      Cliquez sur **OK** pour fermer l’éditeur du schéma et acceptez la propagation proposée par la boîte de dialogue.

4. Dans la table **Attribute table**, renseignez chaque colonne **Value** avec le nom de colonne correspondant dans la table du serveur MS SQL contenant les informations personnelles.
Configurer les composants de sortie

Procédure

1. Double-cliquez sur le premier **tLogRow** pour ouvrir sa vue **Basic settings**.

Dans la zone **Mode**, sélectionnez **Vertical (each row is a key/value list)** pour un affichage optimal des résultats.

2. Configurez de la même manière le second **tLogRow**.

Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder votre Job.
2. Appuyez sur **F6** afin de l'exécuter.

Les informations personnelles contenant l'ID 4 dans la table MS SQL sont affichées dans la table inférieure de la console. Notez que les valeurs des colonnes *id* et *name* affichées dans la table supérieure sont les données d'entrée.
Scénarios associés

Pour des scénarios associés, consultez :

- Scénario : Utiliser le tMssqlSP pour trouver le libellé State à l’aide d’une procédure stockée à la page 2734.

- Scénario : Vérifier le format de numéros à l’aide d’une procédure stockée à la page 2976.

- Scénario : Exécuter une procédure stockée à l’aide du tMDMSP à la page 2320.

Consultez également Scénario : Insérer des données dans des tables mère/fille à la page 2620 si vous voulez analyser un ensemble d’enregistrements d’une table de données ou d’une requête SQL.
tMSSqlTableList

Ce composant liste les noms des tables MS SQL à l’aide d’une instruction SELECT se basant sur une clause WHERE.

Propriétés du tMSSqlTableList Standard

Ces propriétés sont utilisées pour configurer le tMSSqlTableList s’exécutant dans le framework de Jobs Standard.

Le composant tMSSqlTableList Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tMSSqlConnection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Where clause for table name selection</td>
<td>Saisissez la commande WHERE permettant d’identifier les tables sur lesquelles effectuer l’opération d’itération.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>CURRENT_TABLE : nom de la table sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_TABLE : nombre de tables itérées jusqu’à présent. Cette variable est une variable Flow et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le [Guide utilisateur du Studio Talend](#).

---

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Il faut utiliser ce composant en association avec les autres composants MS SQL, notamment avec le <a href="#">tMSSqlConnection</a>.</th>
</tr>
</thead>
</table>

| Dynamic settings | Cliquez sur le bouton [*] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez [Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte](#) à la page 2641 et [Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement](#) à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le [Guide utilisateur du Studio Talend](#). |

---

### Scénario associé

Pour un scénario associé, consultez [Scénario : Itérer une table de base de données et lister le nom des colonnes de la table](#) à la page 2611.
**tMysqlBulkExec**

Ce composant améliore les performances pendant les opérations d’Insert dans une base de données MySQL.

Les composants tMysqlOutputBulk et tMysqlBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tMysqlOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

**Propriétés du tMysqlBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tMysqlBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tMysqlBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>DB Version</strong></td>
<td>Sélectionnez la version de MySQL que vous utilisez.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

| **Host** | Adresse IP du serveur de base de données. |
|**Port** | Numéro du port d’écoute du serveur de base de données. |
|**Database** | Nom de la base de données. |
|**Username et Password** | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
|**Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :

- **None** : n'effectuer aucune action sur la table.
- **Drop and create the table** : supprimer la table puis en créer une nouvelle.
- **Create a table** : créer une table qui n'existe pas encore.
- **Create table if doesn’t exist** : créer la table si nécessaire.
- **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer un nouvelle.
- **Clear a table** : supprimer le contenu de la table.
- **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
|**Table** | Nom de la table à écrire.  
**Remarque :** Une seule table peut être écrite à la fois et cette table doit déjà exister pour que l’opération d’Insert soit autorisée. |
|**Local file Name** | Nom du fichier à charger.  
Ce fichier doit se situer sur la même machine que le **Studio Talend** ou que le Job contenant un **tMysqlBulkExec**, afin que le composant fonctionne correctement. |
### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**. |
| Repository | Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**. |

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Advanced settings

| Additional JDBC parameters | Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. |
| Lines terminated by | Caractère ou suite de caractères utilisés pour séparer les lignes. |
| Fields terminated by | Caractère, chaîne ou expression régulière séparant les champs. |
| Enclosed by | Caractères entourant les données. |

**Action on data**

Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

- **Insert records in table**: Insère de nouveaux enregistrements dans une table.
### tMysqlBulkExec

<table>
<thead>
<tr>
<th>Replace records in table</th>
<th>Remplace les enregistrements existants par de nouveaux.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update records in table</td>
<td>Modifie les enregistrements existants.</td>
</tr>
<tr>
<td>Ignore records in table</td>
<td>Ignore les enregistrements existants ou insère les nouveaux.</td>
</tr>
</tbody>
</table>

**Records contain NULL value**

Cochez cette case si vous souhaitez récupérer les valeurs nulles du flux d’entrée. Si vous ne cochez pas cette case, les valeurs nulles en entrée seront considérées comme des champs vides dans le flux de sortie.

**Encoding**

Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé avec un composant tMysqlOutputBulk. Ensemble, ils offrent un gain de performance important pour l’alimentation d’une base de données MySQL.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Limitation


Scénarios associés

Pour plus d’informations relatives au fonctionnement de tMysqlBulkExec, consultez les scénarios suivants dans :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du composant tMysqlOutputBulk.
- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du composant tMysqlOutputBulkExec.
- Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du composant tOracleBulkExec.
tMysqlClose

Ce composant ferme une connexion à la base de données MySql connectée.

Propriétés du tMysqlClose Standard

Ces propriétés sont utilisées pour configurer le tMysqlClose s’exécutant dans le framework de Jobs Standard.

Le composant tMysqlClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tMysqlConnection dans la liste.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant doit être utilisé avec des composants MySQL, notamment avec le tMysqlCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Ce composant est étroitement lié aux composants tMysqlConnection et tMysqlRollback. Il est généralement utilisé avec un composant tMysqlConnection car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant tMysqlClose, consultez tMysqlConnection à la page 2618.
tMysqlColumnList

Ce composant effectue une opération d’itération sur toutes les colonnes d’une table MySQL donnée et liste les noms des colonnes.

Propriétés du tMysqlColumnList Standard

Ces propriétés sont utilisées pour configurer le tMysqlColumnList s’exécutant dans le framework de Jobs Standard.

Le composant tMysqlColumnList Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tMysqlConnection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Table name</td>
<td>Saisissez le nom de la table.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics      | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>COLUMN_NAME : nom de la colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA_TYPE : type de données de la colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>COLUMN_DEFAULT : valeur par défaut de la colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>IS_NULLABLE : nullabilité de la colonne sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>
retourne **YES** si la colonne autorise les valeurs NULL.
Sinon, elle retourne **NO**.

**COLUMN_KEY** : indicateur clé de la colonne sur laquelle se fait l’itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères si la colonne est configurée à Key. Sinon, elle ne retourne rien.

**CHARACTER_MAXIMUM_LENGTH** : nombre maximum de caractères de la colonne sur laquelle se fait l’itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques pour les types de données binaires ou caractères et retourne NULL pour les autres types.

**NUMERIC_PRECISION** : précision des données de la colonne sur laquelle se fait l’itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques pour les données numériques approximatives, exactes, les entiers, les données monétaires et retourne NULL pour les autres types de données.

**NUMERIC_SCALE** : l’échelle en chiffres autorisée sur la colonne sur laquelle l’itération s’effectue. Cette variable est une variable **Flow** et retourne une chaîne de caractères numériques pour les données numériques approximatives, exactes, les entiers, les données monétaires et retourne NULL pour les autres types de données.

**NB_COLUMN** : nombre de colonnes itérées jusqu’à présent. Cette variable est une variable **After** et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Il faut utiliser ce composant en association avec les autres composants MySQL, notamment avec le <em>tMysqlConnection</em>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre co</td>
</tr>
</tbody>
</table>
nnexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


---

**Scénario : Itérer une table de base de données et lister le nom des colonnes de la table**

Le scénario Java suivant permet de créer un Job à cinq composants. Ce Job permet d’itérer une table donnée à partir d’une base de données MySQL en utilisant la commande Where et de lister tous les libellés des colonnes de la table.

- Dans la Palette, cliquez sur chacun des composants suivants et déposez-les dans l’espace de modélisation : tMysqlConnection, tMysqlTableList, tMysqlColumnList, tFixedFlowInput et tLogRow.
- Reliez le tMysqlConnection au tMysqlTableList par le biais d’un lien OnSubjobOk.
- Reliez les composants tMysqlTableList, tMysqlColumnList et tFixedFlowInput à l’aide de liens Iterate.
- Reliez le tFixedFlowInput au tLogRow par le biais d’un lien Row Main.
Dans l'espace de modélisation, sélectionnez le composant tMysqlConnection et cliquez sur la vue Component pour définir les paramètres de base.

Dans l'onglet Basic settings, entrez manuellement les propriétés de connexion à la base de données ou sélectionnez-les à partir de la liste des variables de contexte : appuyez simultanément sur les touches Ctrl+Espace puis cliquez sur le champ correspondant si vous avez stocké les informations en local sous le nœud DB Connections du répertoire Metadata.

Pour plus d'informations sur les métadonnées, veuillez consulter le Guide utilisateur du Studio Talend.

Dans cet exemple, connectez-vous à une base de données MySQL appelée customers.

Dans l'espace de modélisation, sélectionnez le composant tMysqlTableList et cliquez sur la vue Component pour définir ses paramètres de base.

Dans le champ Component list, sélectionnez le composant de connexion MySQL adéquat si vous utilisez plus d'une connexion.

Renseignez la clause Where en utilisant la bonne syntaxe dans le champ correspondant pour itérer les noms de la ou des tables que vous voulez lister dans la console.

Dans ce scénario, itérez la table appelée customer.

Dans l'espace de modélisation, sélectionnez le composant tMysqlColumnList et cliquez sur la vue Component pour paramétrer ses propriétés de base.

Dans le champ Component list, sélectionnez le composant de connexion MySQL adéquat si vous utilisez plus d'une connexion.
• Dans le champ **Table name**, saisissez le nom de la table dont vous souhaitez lister les libellés de colonnes.
  
  Dans ce scénario, l’objectif est de lister les libellés des colonnes de la table **customer**.

• Dans l’espace de modélisation, sélectionnez le composant **tFixedFlowInput** et cliquez sur la vue **Component** pour paramétrer ses propriétés de base.

• Dans le champ **Schema**, sélectionnez l’option **Built-In** et cliquez sur le bouton […] à côté du champ **Edit Schema** pour définir les données que vous souhaitez utiliser en entrée. Dans ce scénario, le schéma est composé de deux colonnes : la première **TableName** correspond au nom de la table MySQL et la deuxième **ColumnName** correspond aux libellés des colonnes.

  ![Schema of tFixedFlowInput_1](image)

• Cliquez sur **OK** pour fermer la boîte de dialogue et une nouvelle boîte de dialogue apparaît vous demandant si vous souhaitez propager ces modifications, acceptez en cliquant sur **Yes**. Les colonnes définies apparaissent dans le champ **Values** de la vue **Basic settings**.

• Dans le champ **Values**, cliquez sur chacune des cellules de la colonne **Value** pour définir le contenu des colonnes de la table MySQL et appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales.

• Dans la liste des variables globales, sélectionnez : 

  ```
  ((String)globalMap.get("tMysqlTableList_1_CURRENT_TABLE"))
  ```

  pour la colonne **TableName** et

  ```
  ((String)globalMap.get("tMysqlColumnList_1_COLUMN_NAME"))
  ```

  pour la colonne **ColumnName**.

• Dans l’espace de modélisation, sélectionnez le composant **tLogRow**.

• Cliquez sur la vue **Component** et paramétrez les propriétés de base du **tLogRow** en fonction de vos besoins.

• Enregistrez votre Job et appuyez sur **F6** pour l’exécuter.
Le nom de la table MySQL et le libellé de ses colonnes sont affichés dans la console.

```
Starting job Column_Table_List at 00:55 17/11/2008
customer|id
customer|First_Name
customer|Last_Name
customer|Address
customer|id_State
Job Column_Table_List ended at 00:55 17/11/2008. [exit code=0]
```
Ce composant commite en une fois en utilisant une connexion unique, une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes et permet donc un gain de performance.

Le tMysqlCommit valide les données traitées dans un Job à partir d’une base de données connectée.

**Propriétés du tMysqlCommit Standard**

Ces propriétés sont utilisées pour configurer le tMysqlCommit s'exécutant dans le framework de Jobs Standard.

Le composant tMysqlCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque** :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S'il y a plus d'une connexion dans le Job en cours, sélectionnez le composant tMysqlConnection dans la liste.</td>
</tr>
</tbody>
</table>
| Close connection | Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. Avertissement :

Si vous utilisez un lien de type Row > Main pour relier le tMysqlCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec des composants MySQL et notamment avec les composants tMysqlConnection et tMysqlRollback. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend. |

Scénario associé

Pour un scénario associé au composant tMysqlCommit, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
**tMysqlConnection**

Ce composant ouvre une connexion à la base de données MySQL spécifiée afin de pouvoir la réutiliser dans le ou les sous-job(s) suivant(s).

**Propriétés du tMysqlConnection Standard**

Ces propriétés sont utilisées pour configurer le tMysqlConnection s'exécutant dans le framework de Jobs Standard.

Le composant tMysqlConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Table Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Use or register a shared DB Connection</td>
<td>Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion.</td>
</tr>
</tbody>
</table>
partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant `tRunJob`. Utiliser une connexion partagée avec un composant `tRunJob` ayant une de ces options activée fera échouer votre Job.

Cette case est indisponible lorsque la case **Specify a data source alias** est cochée.

### Specify a data source alias

Cochez cette case et spécifiez l’alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans *Talend Runtime* pour récupérer les données d’une base de données MySQL à la page 2647.

### Advanced settings

#### Auto Commit

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commit pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

#### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu’au niveau de chaque composant.
**Utilisation**

| Règle d'utilisation | Ce composant est généralement utilisé avec des composants MySQL, notamment les composants tMysqlCommit et tMysqlRollback. |

**Scénario : Insérer des données dans des tables mère/fille**

Le Job suivant est dédié à des utilisateurs avancés de base de données qui souhaitent effectuer des insertions dans des tables multiples à l’aide de l’*id* d’une table parent qui alimentera une table enfant.

Avant de créer ce Job, suivez les étapes décrites ci-dessous pour créer les tables nécessaires, à l’aide d’un gestionnaire de table tel que *innodb*.

1. Dans un éditeur de ligne de commande, connectez-vous à votre serveur MySQL.

2. Une fois connecté à la base de données correspondante, saisissez la commande suivante afin de créer la table parent :
   ```
   create table f1090_mum(id int not null auto_increment, name varchar(10), primary key(id)) engine=innodb;
   ```

3. Puis créez la deuxième table :
   ```
   create table baby (id_baby int not null, years int) engine=innodb;
   ```

Dans le Studio Talend, le Job est constitué de sept composants, notamment un tMysqlConnection et un tMysqlCommit.

**Construire le Job**

**Procédure**

1. Cliquez et déposez les composants suivants de la Palette : un tFileList, un tFileInputDelimited, un tMap, un tMysqlConnection, un tMysqlCommit et deux tMysqlOutput.

2. Connectez le composant tMysqlConnection au tFileList à l’aide d’une connexion de type OnComponentOk.

3. Connectez le composant tFileList au tMysqlCommit à l’aide d’un lien OnComponentOk.

4. Reliez le tFileList au tFileInputDelimited à l’aide d’un lien Iterate. Le nom du fichier à traiter est automatiquement renseigné depuis le répertoire du tFileList, grâce à une variable globale.

5. Reliez le composant tFileInputDelimited au tMap et divisez le flux entre les deux composants Mysql de sortie. Utilisez un lien Row > Main pour chaque connexion, représentant le flux de données principal.
Configurer les composants

Procédure

1. Configurez les propriétés du tFileList, notamment le nom du répertoire duquel les fichiers vont être récupérés.

2. Dans la vue Component du composant tMysqlConnection, définissez les informations de connexion manuellement ou récupérez-les dans le Repository si vous les avez préalablement stockées dans le dossier DB connection du répertoire Metadata. Pour plus d'informations concernant les Métadonnées, consultez le Guide utilisateur du Studio Talend.

3. Dans l’onglet Basic settings du composant tFileInputDelimited, appuyez sur les touches Ctrl + Espace pour accéder à la liste des variables. Renseignez le champ File Name avec la variable globale : tFileList_1.CURRENT_FILEPATH

4. Paramétrez les autres champs comme vous le souhaitez, définissez les séparateurs de lignes et de champs (Row et Field Separator) en fonction de la structure de votre document.

5. Définissez manuellement le schéma du fichier à l’aide du bouton Edit schema ou sélectionnez le schéma dans le Repository. En version Java, assurez-vous que le type de données est correctement défini, conformément à la nature des données traitées.

6. Dans la zone Output du tMap, ajoutez deux tables de sortie, une que vous nommerez mum pour la table parent et une deuxième que vous nommerez baby pour la table enfant.

Cliquez sur la colonne Name de la zone Input et glissez-la dans la table mum.
Cliquez sur la colonne *Years* de la zone **Input** et glissez-la dans la table *baby*.

7. Assurez-vous que la table *mum* est bien au dessus de la table *baby* car l’ordre des tables détermine la séquence des flux et ainsi l’insertion de base de données est exécutée correctement.

Sélectionnez une connexion de sortie de type **Row** pour acheminer correctement le flux vers le composant de sortie correspondant.

8. Dans l’onglet **Basic settings** des deux composants **tMysqlOutput**, cochez la case **Use an existing connection** pour récupérer les informations du composant **tMysqlConnection**.

9. Nommez votre table dans le champ **Table** en vous assurant que vous saisissez un nom pour la bonne table. Dans cet exemple, la table est soit *f1090_mum*, soit *f1090_baby*.

Sélectionnez **Insert** dans le champ **Action on data** pour les deux composants de sortie (*tMysqlOutput*).

Cliquez sur le bouton **Sync columns** pour récupérer le schéma défini dans le **tMap**.

10. Dans la zone **Additional columns** de l’onglet **Advanced settings** du composant **tMysqlOutput** correspondant à la table enfant (*f1090_baby*), définissez la colonne *id_baby* de manière à ce que l’id de la table parent soit réutilisé.

11. Dans le champ **SQL expression**, saisissez `'(Select Last_Insert_id())'`

Dans la colonne **Position**, sélectionnez l’option **Before** et dans **Reference column**, l’option *years*.

Dans l’onglet **Advanced settings**, décochez la case **Extend insert**.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour enregistrer votre Job.
2. Appuyez sur F6 pour l'exécuter.

**Résultats**

L'id de la table parent a été réutilisé pour alimenter la colonne id_baby.

**Scénario : Annuler l'insertion de données dans des tables mère/fille**

**Pourquoi et quand exécuter cette tâche**

En vous basant sur Scénario : Insérer des données dans des tables mère/fille à la page 2620 du composant tMysqlConnection, insérez une opération de rollback (annulation) afin d'éviter un commit inutile.

**Procédure**

1. Glissez-déposez un tMysqlRollback de la Palette dans l'espace de modélisation graphique. Le tMysqlRollback assure que la transaction ne sera pas commitée partiellement.
2. Connectez le tFileList au tMysqlRollback à l'aide d’un lien OnSubjobError.
3. Double-cliquez sur le tMysqlRollback afin d’ouvrir sa vue Basic settings.
4. Dans la liste Component List, sélectionnez le composant de connexion à utiliser.
Scénario : Partager une connexion de base de données entre un Job père et un Job fils

Ce scénario montre comment partager une connexion entre un Job père et un Job fils. Le Job père appelle le Job fils afin d'écrire des données aléatoires dans une base de données MySQL. Il lit ensuite les données depuis la base de données MySQL puis les affiche dans la console. La connexion à la base de données MySQL est configurée une seule fois et utilisée dans les deux Jobs.

Configurer le Job fils

Ajouter et relier les composants

Procédure

1. Ajoutez les composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la Palette :
   • un tMysqlConnection, afin d'ouvrir une connexion à la base de données MySQL,
   • un tRowGenerator, afin de générer des données d'entrée aléatoires,
   • un tMysqlOutput, afin d'écrire des données dans la base de données MySQL.
2. Reliez le tRowGenerator au tMysqlOutput à l'aide d'un lien Row > Main.
3. Reliez le tMysqlConnection au tRowGenerator à l'aide d'un lien Trigger > OnSubjobOk.

Configurer la connexion à la base de données

Procédure

1. Double-cliquez sur le tMysqlConnection afin d'ouvrir sa vue Basic settings.
2. Dans la liste **Property Type**, sélectionnez **Built-In** puis configurez les détails de connexion dans les champs appropriés, dont :

- le nom d’hôte ou l’adresse IP de votre serveur de base de données (**Host**),
- le numéro du port d’écoute (**Port**),
- le nom de la base de données (**Database**),
- le nom d’utilisateur (**Username**) et le mot de passe (**Password**) pour l’identification à la base de données.

Si vous avez stocké les informations de connexion sous le nœud **Metadata** du **Repository**, vous pouvez déposer ces métadonnées centralisées sur le composant tMysqlConnection. Pour plus d’informations concernant la centralisation d’une connexion à une base de données, consultez le chapitre sur la gestion des métadonnées dans le Guide utilisateur du Studio Talend.

3. Cochez la case **Use or register a shared DB Connection** afin que la connexion à la base de données ouverte par ce composant puisse être partagée entre les différents Jobs. Ensuite, dans le champ **Shared DB Connection Name**, saisissez un nom pour la connexion partagée, entre guillemets doubles, **shared_mysql_connection** dans cet exemple.

**Configurer les données d’entrée**

**Procédure**

1. Double-cliquez sur le composant **tRowGenerator** afin d’ouvrir l’éditeur **Row Generator**.
2. Cliquez sur le bouton [+] afin d’ajouter quatre colonnes puis définissez leurs propriétés comme suit :
   - *id*, de type *Integer* et d’une longueur de 2 caractères,
   - *firstName*, de type *String* et d’une longueur de 15 caractères,
   - *lastName*, de type *String* et d’une longueur de 15 caractères,
   - *city*, de type *String* et d’une longueur de 15 caractères.
3. Définissez la fonction de chaque colonne :
   - Pour la colonne *id*, sélectionnez *Numeric.sequence* dans la liste *Function* afin de générer des séquences de nombres.
   - Pour la colonne *firstName*, sélectionnez *TalendDataGenerator.getFirstName* dans la liste *Function* afin de générer des prénoms de manière aléatoire.
   - Pour la colonne *lastName*, sélectionnez *TalendDataGenerator.getLastName* dans la liste *Function* pour générer des noms de famille de manière aléatoire.
   - Pour la colonne *city*, sélectionnez *TalendDataGenerator.getUsCity* dans la liste *Function* pour générer des noms de ville de manière aléatoire.
4. Dans le champ *Number of Rows for RowGenerator*, précisez le nombre de colonnes de données que vous souhaitez générer, 10 dans cet exemple.
5. Cliquez sur le bouton *Preview* de l’onglet *Preview* afin de valider les paramètres du générateur.
6. Une fois les paramètres validés, cliquez sur *OK* afin de fermer l’éditeur. Dans la boîte de dialogue qui s’affiche, cliquez sur *Yes* afin de propager les changements au composant suivant.
Configurer la sortie de la base de données

Procédure

1. Double-cliquez sur le tMysqlOutput afin d’ouvrir sa vue Basic settings.

![Image du tMysqlOutput]

2. Cochez la case Use an existing connection et, si vous avez plus d’un composant de connexion dans ce Job, sélectionnez le composant de connexion à utiliser dans la liste Component List.

3. Dans le champ Table, saisissez le nom de la table de base de données dans laquelle les données sont écrites, customers dans cet exemple.

4. Dans la liste Action on table, sélectionnez l’option Drop table if exists and create afin de vous assurer de la création d’une table vide.

5. Dans la liste Action on data, sélectionnez Insert.

6. Cliquez sur le bouton [...] à côté du champ Edit schema afin de vérifier la propagation du schéma de sortie. Si besoin, cliquez sur Sync columns afin de récupérer le schéma depuis le composant précédent.

7. Appuyez sur Ctrl+S afin de sauvegarder votre Job.

Configurer le Job père

Construire le flux de données dans le Job parent

Procédure

1. Ajoutez les composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette :
   - un tRunJob, afin d’appeler le Job fils,
   - un tMysqlConnection, afin d’ouvrir la connexion à la base de données MySQL,
   - un tMysqlInput, afin de lire les données écrites dans la base de données MySQL par le Job fils,
   - un tLogRow, afin d’afficher les données dans la console,
   - un tMysqlCommit, afin de commiter les données lors de la transaction et fermer la connexion à la base de données.

2. Reliez le tRunJob au tMysqlConnection à l’aide d’un lien Trigger > OnSubjobOk.

3. Reliez le tMysqlConnection au tMysqlInput à l’aide d’un lien Trigger > OnSubjobOk.

4. Reliez le tMysqlInput au tLogRow à l’aide d’un lien Row > Main.

5. Reliez le tMysqlInput au tMysqlCommit à l’aide d’un lien Trigger > OnSubjobOk.
Configurer les composants

Procédure

1. Double-cliquez sur le `tRunJob` afin d’ouvrir sa vue `Basic settings`. Si un Job fils a déjà été défini pour le composant, vous pouvez cliquer-droit sur le composant et sélectionner `Settings` dans le menu contextuel afin d’ouvrir la vue `Basic settings`. Vous pouvez également cliquer sur le composant et sélectionner l’onglet `Component`.


3. Dans la vue `Basic settings` du composant `tMysqlConnection`, cochez la case `Use or register a shared DB Connection`. Ensuite, dans le champ `Shared DB Connection Name`, saisissez le nom de la connexion partagée définie dans le Job fils, `shared_mysql_connection` dans cet exemple. Laissez les autres paramètres vides ou tels qu’ils sont.
4. Double-cliquez sur le MySQLInput afin d’ouvrir sa vue Basic settings.

5. Cochez la case Use an existing connection et, dans le cas où vous avez plus d’un composant de connexion dans ce Job, sélectionnez le composant de connexion à utiliser dans la liste Component List.

Une fois la structure de données définie, cliquez sur OK afin de fermer l'éditeur de schéma. Dans la boîte de dialogue qui s'affiche, cliquez sur Yes afin de propager les changements au composant suivant.

7. Précisez le nom de la tâche dans la liste Table Name et cliquez sur Guess Query afin de renseigner automatiquement le champ Query avec la requête appropriée.

8. Dans la vue Basic settings du tLogRow, sélectionnez le mode Table afin d'afficher le résultat de l'exécution sous forme de tableau. Laissez les autres paramètres du tMysqlCommit tels qu'ils sont.

Exécuter le Job

Procédure

1. Appuyez sur Ctrl+S afin de sauvegarder votre Job.

2. Appuyez sur F6 ou cliquez sur le bouton Run de la console Run afin d'exécuter votre Job.

Le Job père appelle le Job fils afin d'écrire des données dans la base de données. Il lit ensuite les données depuis la base de données puis les affiche dans la console.
tMysqlInput

Ce composant exécute une requête de base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma.

Le tMysqlInput lit une base de données et en extrait des champs à l’aide de requêtes. Les champs récupérés sont ensuite transmis au composant suivant via une connexion de flux (Main row).

Propriétés du tMysqlInput Standard

Ces propriétés sont utilisées pour configurer le tMysqlInput s’exécutant dans le framework de Jobs Standard.

Le composant tMysqlInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-In ou Repository.</td>
</tr>
<tr>
<td></td>
<td>Built-In : propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>DB version</td>
<td>Sélectionnez la version de la base de données à utiliser.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une
connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs.</td>
</tr>
</tbody>
</table>

**Built-in** : Le schéma sera créé et conservé pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

**Repository** : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez
propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Nom de la table à lire.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query type et Query</strong></td>
<td>Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs correspond à celui défini dans le schéma.</td>
</tr>
<tr>
<td><strong>Specify a data source alias</strong></td>
<td>Cochez cette case et spécifiez l’alias de la source de données créée dans <em>Talend Runtime</em> pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans <em>Talend Runtime</em>. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans <em>Talend Runtime</em> pour récupérer les données d’une base de données MySQL à la page 2647. <strong>Avertissement</strong> : Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l’alias de la source de données spécifié. Cette option est indisponible lorsque la case <strong>Use an existing connection</strong> est cochée.</td>
</tr>
</tbody>
</table>

### Advanced settings

| Additional JDBC parameters | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**. **Remarque** : Lorsque vous devez traiter des données au format date/heure *0000-00-00 00:00:00* utilisant ce composant, définissez les paramètres comme suit : noDateTimeStringSync=true&zeroDateTimeBehavior=convertToNull. |
| Enable stream | Cochez cette case pour déterminer avec quelles lignes vous souhaitez travailler. Cette option permet d’améliorer les performances. Cette case est disponible uniquement lorsque l’option **Mysql 4** ou **Mysql 5** est sélectionnée dans la liste **DB Version**. |
### Trim all the String/Char columns

Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.

### Trim column

Supprimez les espaces en début et en fin de champ dans les colonnes sélectionnées.

**Remarque :**

Décochez Trim all the String/Char columns pour activer le tableau Trim column.

### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

---

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

---

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données MySQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario : Ecrire des colonnes d’une base de données MySQL dans un fichier de sortie en utilisant tMysqlInput**

Dans ce scénario, vous allez lire des colonnes d’une base de données MySQL puis les écrire dans une table d’un fichier local de sortie.

**Déposer et relier les composants**

**Procédure**

1. Déposez un composant tMysqlInput et un tFileOutputDelimited de la Palette dans l’espace de modélisation graphique.
2. Reliez le tMysqlInput au tFileOutputDelimited à l’aide d’un lien Row > Main.

![Diagramme de connexion](image)

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le composant tMysqlInput afin d’ouvrir l’onglet Basic settings de la vue Component.
2. Dans la liste **Property Type**, sélectionnez **Repository**, si vous avez déjà stocké la connexion à la base de données dans le nœud **Metadata** du **Repository**. Les informations de connexion sont alors automatiquement renseignées.

Pour plus d'informations concernant le stockage des métadonnées dans le **Repository**, consultez le **Guide utilisateur du Studio Talend**.

Si vous n'avez pas stocké localement la connexion dans le **Repository**, sélectionnez **Built-In** dans la liste **Property Type** puis renseignez manuellement les informations.

3. Dans la liste **Schema**, sélectionnez **Built-In** puis cliquez sur **Edit schema** pour configurer le schéma.

L'éditeur du schéma s'ouvre :

4. Cliquez quatre fois sur le bouton pour ajouter quatre lignes au schéma.

Dans les colonnes **Column** et **Db Column**, cliquez dans les champs afin de saisir le nom des colonnes correspondantes, *id*, *first_name*, *city* et *salary*, respectivement.

Cliquez sur les champs de la colonne **Type** pour définir le type de données.

Cliquez sur **OK** pour fermer l'éditeur du schéma.

5. A côté du champ **Table Name**, cliquez sur le bouton [...] pour sélectionner la table de la base de données souhaitée.
Une boîte de dialogue affiche une arborescence de toutes les tables dans la base de données sélectionnée :

6. Cliquez sur la table qui vous intéresse puis sur OK pour fermer la boîte de dialogue.

7. Dans la liste Query Type, sélectionnez Built-In.

8. Dans le champ Query, saisissez la requête permettant de récupérer les colonnes souhaitées de la table.

```
"SELECT
  "employees"."id",
  "employees"."first_name",
  "employees"."city",
  "employees"."salary"
FROM "employees"
```


10. A côté du champ File Name, cliquez sur le bouton [...] afin de parcourir votre système jusqu’à l’emplacement où vous souhaitez sauvegarder le fichier de sortie, puis saisissez un nom pour le fichier.

   Cochez la case Include Header afin de récupérer le nom des colonnes du schéma ainsi que les données.

11. Sauvegardez le Job en appuyant sur les touches Ctrl+S.

**Exécuter le Job**

Appuyez sur F6 pour exécuter votre Job.
Dans le fichier de sortie est écrit le nom des colonnes souhaitées et les données correspondantes, récupérés de la base de données :

**Remarque :**
Le Job peut également être exécuté en mode **Traces Debug**, ce qui vous permet de voir les lignes pendant leur écriture dans le fichier de sortie, dans l'espace de modélisation graphique.

**Scénario : Utiliser des paramètres de contexte lors de la lecture d'une table d'une base de données**

Dans ce scénario, MySQL est utilisé à titre d'exemple. Une table d'une base de données MySQL est lue, à l'aide un paramètre de contexte référant au nom de la table.

**Déposer et relier les composants**

**Procédure**

1. Déposez un composant **tMysqlInput** et un **tLogRow** de la Palette dans l'espace de modélisation graphique.
2. Reliez le **tMysqlInput** au **tLogRow** à l'aide d'un lien **Row > Main**.

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le **tMysqlInput** pour ouvrir la vue **Basic settings**.
2. Dans la liste Property Type, sélectionnez Repository si vous avez déjà stocké la connexion à la base de données dans le nœud Metadata du Repository. Les champs des propriétés sont automatiquement renseignés.

Pour plus d’informations concernant le stockage d’une connexion à une base de données, consultez le Guide utilisateur du Studio Talend.

Si vous n’avez pas configuré de connexion à une base de données dans le Repository, renseignez manuellement les informations après avoir sélectionné Built-in dans la liste Property Type.

3. Configurez le Schema en Built-In, puis cliquez sur Edit schema pour définir le schéma comme vous le souhaitez.

L’éditeur du schéma s’ouvre :

![éditeur du schéma]

4. Cliquez sur le bouton pour ajouter les sept colonnes à utiliser pour configurer le schéma et renommez-les respectivement : id, first_name, last_name, city, state, date_of_birth et salary.

Cliquez sur les lignes de la colonne Type pour définir le type des données. 

Cliquez sur OK pour fermer l’éditeur du schéma.

5. Placez votre souris sur le champ Table Name et appuyez sur F5 pour configurer les paramètres de contexte.
Pour plus d’informations concernant les paramètres de contexte, consultez le Guide utilisateur du Studio Talend.

6. Laissez les paramètres par défaut dans le champ **Name** et saisissez le nom de la table de la base de données dans le champ **Default value**, *employees* dans ce cas.

7. Cliquez sur **Finish** pour valider.

Le paramètre de contexte *context.TABLE* apparaît automatiquement dans le champ **Table Name**.

8. Dans la liste **Query type**, sélectionnez **Built-In** puis cliquez sur **Guess Query** pour obtenir l'instruction de la requête.

Dans ce scénario, lisez les enregistrements contenant un salaire supérieur à 8000. Ajoutez une clause *Where*. L'instruction finale s’affiche comme suit :

```
"SELECT "+context.TABLE+.\'id\', "+context.TABLE+.\'first_name\', "+context.TABLE+.\'last_name\', "+context.TABLE+.\'city\', "+context.TABLE+.\'state\', "+context.TABLE+.\'date_of_birth\', "+context.TABLE+.\'salary"
FROM "+context.TABLE+
WHERE
"+context.TABLE+.\'salary’ > 8000"
```

9. Double-cliquez sur le **tLogRow** pour configurer ses propriétés de base dans l’onglet **Basic settings**.
Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un meilleur affichage des résultats.

**Sauvegardez le Job.**

**Exécuter le Job**

Appuyez sur **F6** pour exécuter votre Job. Les résultats s'affichent dans la console.

```
<table>
<thead>
<tr>
<th>id</th>
<th>first_name</th>
<th>last_name</th>
<th>city</th>
<th>state</th>
<th>date_of_birth</th>
<th>salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Martin</td>
<td>Roosevelt</td>
<td>Sacramento</td>
<td>MD</td>
<td>19-10-1950</td>
<td>9011</td>
</tr>
<tr>
<td>2</td>
<td>Zachary</td>
<td>Johnson</td>
<td>Atlanta</td>
<td>MD</td>
<td>14-11-1978</td>
<td>8118</td>
</tr>
<tr>
<td>4</td>
<td>Herbert</td>
<td>Harrison</td>
<td>Charleston</td>
<td>AB</td>
<td>15-08-1963</td>
<td>9233</td>
</tr>
</tbody>
</table>
```

Comme vous pouvez le constater ci-dessus, les enregistrements comportant un salaire supérieur à 8000 sont récupérés.

**Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte**

Dans ce scénario, MySQL est utilisé à titre d'exemple. Vous allez lire des données depuis des tables de bases de données ayant la même structure mais se trouvant dans deux différentes bases de données MySQL nommées respectivement **project_q1** et **project_q2**. Vous allez spécifier dynamiquement les connexions à ces bases de données au moment de l'exécution, sans modifier le Job.

**Déposer et relier les composants**

**Procédure**

1. Déposez deux composants **tMysqlConnection**, un **tMysqlInput**, un **tLogRow** et un **tMysqlClose** dans l'espace de modélisation graphique.
2. Reliez le premier **tMysqlConnection** au second **tMysqlConnection** et le second **tMysqlConnection** au **tMysqlInput** à l'aide de liens **Trigger > On Subjob Ok**.
3. Reliez le **tMysqlInput** au **tLogRow** à l'aide d'un lien **Row > Main**.
4. Reliez le **tMysqlInput** au **tMysqlClose** à l'aide d'un lien **Trigger > On Subjob Ok**.
 Créer une variable de contexte

Pourquoi et quand exécuter cette tâche

Afin de pouvoir choisir dynamiquement une connexion à la base de données au moment de l’exécution, vous devez définir une variable de contexte. Cette dernière est configurée dans les paramètres Dynamic settings du composant d’entrée de la base de données.

Procédure

1. Dans la vue Contexts, cliquez sur le bouton [+] afin d’ajouter une ligne à la table, puis cliquez dans le champ Name et saisissez un nom pour la variable, myConnection dans cet exemple.

2. Dans le champ de la liste Type, sélectionnez List Of Value.

3. Cliquez dans le champ Value, puis cliquez sur le bouton qui apparaît dans ce champ pour ouvrir la boîte de dialogue [Configure value of list].
4. Dans la boîte de dialogue [Configure value of list], cliquez sur le bouton New... pour ouvrir la boîte de dialogue [New Value] et saisissez le nom de l’un des composants de connexion dans le champ de texte, tMysqlConnection_1 dans cet exemple. Cliquez ensuite sur OK pour fermer la boîte de dialogue.

Répétez cette étape pour spécifier le nom de l’autre composant de connexion comme autre élément de liste, tMysqlConnection_2 dans cet exemple.

Lorsque vous avez terminé, cliquez sur OK pour fermer la boîte de dialogue [Configure Values].

5. Cochez la case à côté du champ de valeur de la variable et saisissez, dans le champ Prompt, le message que vous souhaitez afficher à l’exécution, Select a connection component: dans cet exemple.
Configurer les composants

Procédure

1. Double-cliquez sur le premier composant **tMysqlConnection** afin d'afficher sa vue **Basic settings** et saisissez les informations de connexion. Pour plus d'informations concernant la configuration du **tMysqlConnection**, consultez **tMysqlConnection** à la page 2618. Noter que ce composant sert à ouvrir une connexion à une base de données MySQL nommée **project_q1**.

   ![tMysqlConnection_1](image)

2. Configurez le second composant **tMysqlConnection** de la même façon mais en saisissant **project_q2** dans le champ **Database**. En effet, ce composant sert à ouvrir une connexion à une autre base de données MySQL, **project_q2**.

   ![tMysqlConnection_2](image)

3. Double-cliquez sur le **tMysqlInput** afin d'ouvrir sa vue **Basic settings**.

   ![tMysqlInput_1](image)
4. Cochez la case **Use an existing connection** et laissez la case **Component List** comme elle est.

5. Cliquez sur le bouton [...] situé à côté du champ **Edit schema** afin d’ouvrir la boîte de dialogue [Schema]. Définissez la structure de la table de la base de données à partir de laquelle les données sont lues.

Dans cet exemple, la structure de la table de la base de données est composée de quatre colonnes : *id* (de type **Integer** et d’une longueur de 2 caractères), *firstName* (de type **String** et d’une longueur de 15 caractères), *lastName* (de type **String** et d’une longueur de 15 caractères) et *city* (de type **String** et d’une longueur de 15 caractères). Cliquez sur **OK** pour fermer la boîte de dialogue et propager le schéma au composant suivant.

6. Dans le champ **Table**, saisissez le nom de la table de la base de données, *customers* dans cet exemple. Cliquez sur **Guess Query** afin de générer l’instruction de la requête correspondant au schéma de votre table dans le champ **Query**.

7. Dans la vue **Dynamic settings**, cliquez sur le bouton [+] afin d’ajouter une ligne à votre table. Dans le champ **Code**, saisissez le code du script de la variable de contexte que vous venez de créer, " + context.myConnection + " dans cet exemple.

8. Dans la vue **Basic settings** du **tLogRow**, sélectionnez l’option **Table** pour un meilleur affichage des résultats d’exécution du Job.
Dans la vue Dynamic settings du tMysqlClose, procédez de la même manière que dans la vue Dynamic settings du tMysqlInput.

Sauvegarder et exécuter le Job

Procédure

   Une boîte de dialogue s’affiche vous demander de spécifier le composant de connexion que vous souhaitez utiliser.
2. Sélectionnez le composant de connexion tMysqlConnection_1 et cliquez sur OK.

Les données lues depuis la base de données project_q1 s’affichent dans la console Run.
3. Appuyez sur F6 ou cliquez sur Run afin d’exécuter le Job à nouveau. Quand la boîte de dialogue s’affiche, sélectionnez l’autre composant de connexion, à savoir tMysqlConnection_2, afin de lire les données depuis l’autre base de données, project_q2.

Les données lues depuis la base de données project_q2 s’affichent dans la console Run.

Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL

Ce scénario décrit un Job à deux composants qui récupère les données d’une table de base de données MySQL et les affiche dans la console. Ce Job va être déployé dans Talend Runtime et va utiliser les données source créées dans le Talend Runtime pour se connecter à la base de données définie dans un fichier de configuration du Runtime.
Créer une source de données MySQL dans le conteneur de Talend Runtime

Afin de pouvoir récupérer les données d’une table de base de données MySQL et les afficher dans la console, vous devez avoir créé une source de données MySQL doit avoir été créée dans le conteneur de Talend Runtime.

Avant de commencer


Procédure

1. Installez le pilote JDBC MySQL en exécutant la commande bundle:install dans le conteneur de Talend Runtime :

   karaf@trun()> bundle:install mvn:mysql/mysql-connector-java/5.1.18

2. Installez le pool de connexions à la base de données en exécutant la commande bundle:install dans le conteneur de Talend Runtime :

   karaf@trun()> bundle:install -s mvn:commons-dbcp/commons-dbcp/1.4

3. Copiez le fichier de configuration de la source de données datasource-mysql.xml du dossier <TalendRuntimePath>/add-ons/datasources/dataservice vers le dossier <TalendRuntimePath>/container/deploy puis modifiez dans ce fichier la valeur des propriétés du nom d’utilisateur et du mot de passe requises pour se connecter à votre base de données.

   ```xml
   <?xml version="1.0" encoding="UTF-8"?>
   <blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
   default-activation="lazy">

   <bean id="mysqlDataSource" class="com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource">
     <property name="url" value="jdbc:mysql://localhost:3306/test"/>
     <property name="user" value="root"/>
     <property name="password" value="talend"/>
   </bean>

   <bean id="dataSource" class="org.apache.commons.dbcp.datasources.SharedPoolDataSource" destroy-method="close">
     <property name="connectionPoolDataSource" ref="mysqlDataSource"/>
     <property name="maxActive" value="20"/>
     <property name="maxIdle" value="5"/>
     <property name="maxWait" value="-1"/>
   </bean>

   <service ref="dataSource" interface="javax.sql.DataSource">
   </service>

   </blueprint>
   ```
Remarque : La propriété `osgi.jndi.service.name` du fichier de configuration de la source de données définit l’alias de la source de données, `jdbc/sample` dans cet exemple. L’alias va être utilisé dans le composant de base de données pour se connecter à la base de données définie dans le fichier de configuration de la source de données lorsque le Job est déployé dans Talend Runtime.

Pour plus d’informations sur l’utilisation de la source de données dans le conteneur de Talend Runtime Container, consultez le Guide *Talend ESB Container Administration Guide* (en anglais) et le *Guide utilisateur de Talend Open Studio for ESB*.

### Ajouter et relier les composants

**Procédure**

1. Créez un nouveau Job et ajoutez un composant **tMysqlInput** et un composant **tLogRow** en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le composant **tMysqlInput** au **tLogRow** à l’aide d’un lien **Row > Main**.

###Configurer les composants

**Procédure**

1. Double-cliquez sur le **tMysqlInput** pour ouvrir sa vue **Basic settings** dans l’onglet **Component**.

![tMysqlInput_1](image)

2. Dans le champ **Table Name**, saisissez le nom de la table de laquelle les données seront récupérées. Dans cet exemple, il s’agit de **student**.

![Schema of tMysqlInput_1](image)

4. Cliquez sur le bouton Guess Query pour renseigner le champ Query par la requête SQL à utiliser pour récupérer les données de la table spécifiée.

5. Cochez la case Specify a data source alias et dans le champ Data source alias qui s’affiche, saisissez l’alias de la source de données spécifié dans le fichier de configuration de la source de données datasource-mysql.xml côté Talend Runtime. Dans cet exemple, il s’agit de jdbc/sample.

6. Double-cliquez sur le composant tLogRow pour ouvrir sa vue Basic settings.

![tLogRow_1](image)

7. Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour rendre le résultat plus lisible.

**Construire et déployer le Job dans le conteneur de Talend Runtime**

**Procédure**

1. Appuyez sur Ctrl + S pour enregistrer le Job.
2. Démarrez le conteneur de Talend Runtime.
3. Dans la vue Repository de votre Studio Talend, cliquez-droit sur le Job et dans le menu contextuel, sélectionnez Build Job pour ouvrir la boîte de dialogue [Build Job].
4. Sélectionnez OSGI Bundle For ESB dans la liste déroulante Select the build type.


Pour plus d’informations sur la construction d’un Job, consultez le Guide utilisateur du Studio Talend.

6. Cliquez sur Finish pour fermer la boîte de dialogue [Build Job].
Comme vous pouvez le voir dans la console du conteneur de **Talend Runtime**, le Job s’exécute immédiatement après avoir été déployé et les données de la table *student* sont récupérées et affichées dans la console.
**tMysqlLastInsertId**

Ce composant récupère la valeur de la clé primaire du dernier enregistrement inséré dans une table MySQL par un utilisateur.

Le tMysqlLastInsertId récupère le dernier ID inséré à partir d’une connexion MySQL spécifiée.

**Propriétés du tMysqlLastInsertId Standard**

Ces propriétés sont utilisées pour configurer le tMysqlLastInsertId s’exécutant dans le framework de Jobs Standard.

Le composant tMysqlLastInsertId Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs.</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
<tr>
<td>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
<td></td>
</tr>
<tr>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les</td>
<td></td>
</tr>
</tbody>
</table>
modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Component list

Sélectionnez le composant tMysqlConnection dans la liste s’il y a plus d’une connexion dans votre Job.

Advanced settings

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

Global Variables

NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant est un composant intermédiaire.

Avertissement :

Si vous utilisez ce composant avec un tMysqlOutput, vérifiez que la case Extend Insert de l’onglet Advanced settings est bien décochée. En effet, Extend Insert permet de faire un chargement par lot, ainsi seul l’ID de la dernière ligne du dernier lot sera retourné.

Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez...
Scénario : Récupérer les ID des dernières entrées ajoutées avec le tMysqlLastInsertId

Le scénario Java suivant permet d'ouvrir une connexion à une base de données MySQL, d'y écrire des données spécifiées et enfin de récupérer les derniers ID ajoutés à la connexion existante.

- Cliquez sur chacun des composants suivants et déposez-les dans l'éditeur graphique : tMysqlConnection, tMysqlCommit, tFileInputDelimited, tMysqlOutput, tMysqlLastInsertId, et tLogRow.

- Reliez le tMysqlConnection au tFileInputDelimited à l'aide d'un lien OnSubjobOk.

- Reliez le tFileInputDelimited au tMysqlCommit à l'aide d'un lien OnSubjobOk.

- Reliez les trois derniers composants au tFileInputDelimited et entre eux à l'aide de liens Row Main, comme suit :

- Sélectionnez le tMysqlConnection en cliquant dessus à partir de l'éditeur graphique.
• Cliquez sur la vue Component pour en définir la configuration de base (Basic settings).

• Dans l’onglet Basic settings, entrez les paramètres de connexion manuellement ou sélectionnez-les à partir d’une liste de variables de contexte, si vous les avez stockées localement sous le noeud DB connections de la partie Metadata (Méthodonnées) du Repository. Vous accéderez à cette liste en cliquant dans le champ à renseigner puis en appuyant sur Ctrl+Espace. Pour plus d’informations sur les Méthodonnées, consultez le Guide utilisateur du Studio Talend.

• Sélectionnez le tMysqlCommit en cliquant dessus à partir de l’éditeur graphique, puis cliquez sur la vue Component pour en définir la configuration de base (Basic settings).

• Dans le champ Component list, sélectionnez le composant tMysqlConnection approprié si vous effectuez un Job à plusieurs connexions.

• Sélectionnez le tFileInputDelimited en cliquant dessus à partir de l’éditeur graphique.

• Cliquez sur la vue Component pour en définir la configuration de base (Basic settings).

• Renseignez le chemin d’accès au fichier à traiter dans le champ File Name. Dans cet exemple, le nom du fichier est Customers.

• Dans le champ Row separator, définissez le séparateur de lignes permettant d’identifier les fins de ligne. De même, pour Field separator, définissez le séparateur de champs permettant de délimiter les différents champs pour chaque ligne.

• Définissez, si nécessaire, le nombre de lignes d’en-tête et de pied-de-page à ignorer, ainsi que le nombre maximum de lignes que vous souhaitez traiter. Dans cet exemple, le fichier comporte une ligne d’en-tête.

• Cliquez sur le bouton [...] à côté de la mention Edit Schema pour définir les données à passer au composant suivant.
Dans ce scénario, le schéma se compose de deux colonnes, *name* et *age*. La première colonne comprend le nom des employés et la seconde comporte leurs âges respectifs.

- Sélectionnez le `tMysqlOutput` en cliquant dessus à partir de l'éditeur graphique.
- Cliquez sur la vue *Component* pour en définir la configuration de base (*Basic settings)*.

  - Cochez la case *Use an existing connection*.
  - Dans le champ *Table*, renseignez le nom de la table dans laquelle vous voulez écrire la liste des employés. Dans cet exemple, il s'agit de la table *employee*.
  - Sélectionnez les actions à effectuer dans les champs *Action on table* et *Action on data*, à partir des listes proposées. Dans cet exemple, n’effectuez aucune action sur la table elle-même mais exécutez une action *Insert* sur les données.
  - Cliquez sur le bouton *Sync columns* pour synchroniser les colonnes par rapport au composant précédent. Dans cet exemple, le schéma à reproduire dans la table de la base de données MySQL se compose des deux colonnes *name* et *age*.

Sujets associés : consultez le *Guide utilisateur du Studio Talend*. 
• Sélectionnez le **tMysqlLastInsertId** en cliquant dessus à partir de l’éditeur graphique.

• Cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**).

• Dans le champ **Component list**, sélectionnez le composant **tMysqlConnection** approprié si vous effectuez un Job à plusieurs connexions.

• Cliquez sur le bouton **Sync columns** pour synchroniser les colonnes par rapport au composant précédent. Dans le schéma de sortie du **tMysqlLastInsertId**, vous pouvez voir la colonne last_insert_id qui indique les dernières ID ajoutées à la connexion existante.

Vous pouvez sélectionner le type de données **Long** dans la liste déroulante **Type**, dans le cas de nombreuses entrées.

• Sélectionnez le **tLogRow** en cliquant dessus à partir de l’éditeur graphique, puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**). Pour plus d’informations, consultez le **tLogRow** à la page 2105.
• Sauvegardez votre Job puis appuyez sur **F6** pour l’exécuter.

<table>
<thead>
<tr>
<th>Marie</th>
<th>24</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patrick</td>
<td>22</td>
<td>41</td>
</tr>
<tr>
<td>Pierrick</td>
<td>27</td>
<td>42</td>
</tr>
</tbody>
</table>

Le composant **tMysqlLastInsertId** affiche les dernières ID ajoutées pour chaque ligne de la connexion existante.
tMysqlLookupInput

Ce composant lit une base de données MySQL et extrait des champs en se basant sur une requête.

Le tMysqllookupInput exécute une requête de base de données dans un ordre défini correspondant au schéma.

Ce composant vous permet également de vous connecter et de lire des données d'une base de données RDS Aurora ou RDS MySQL.
tMysqlOutput

Ce composant écrit, met à jour, modifie ou supprime les données d'une base de données.
Le tMysqlOutput exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.

Propriétés du tMysqlOutput Standard

Ces propriétés sont utilisées pour configurer le tMysqlOutput s'exécutant dans le framework de Jobs Standard.
Le composant tMysqlOutput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d'un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
</tbody>
</table>
| Property type          | Peut être Built-In ou Repository. 
                         * Built-In : propriétés utilisées ponctuellement. 
                         * Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |
| DB Version             | Sélectionnez la version de MySQL que vous utilisez. |
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Vous ne pouvez écrire qu’une seule table à la fois.</td>
</tr>
<tr>
<td><strong>Action on table</strong></td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td><strong>Default</strong> : n’effectuer aucune opération de table.</td>
</tr>
<tr>
<td></td>
<td><strong>Drop and create the table</strong> : supprimer la table puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td><strong>Create a table</strong> : créer une table qui n’existe pas encore.</td>
</tr>
<tr>
<td></td>
<td><strong>Create table if doesn’t exist</strong> : créer la table si nécessaire.</td>
</tr>
<tr>
<td></td>
<td><strong>Drop a table if exists and create</strong> : supprimer la table si elle existe déjà, puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td><strong>Clear a table</strong> : supprimer le contenu de la table.</td>
</tr>
<tr>
<td></td>
<td><strong>Truncate table</strong> : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.</td>
</tr>
<tr>
<td><strong>Action on data</strong></td>
<td>Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td><strong>Insert</strong> : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.</td>
</tr>
<tr>
<td></td>
<td><strong>Update</strong> : Mettre à jour les entrées existantes.</td>
</tr>
<tr>
<td></td>
<td><strong>Insert or update</strong> : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.</td>
</tr>
</tbody>
</table>
**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

**Delete** : Supprimer les entrées correspondantes au flux d’entrée.

**Insert or update on duplicate key or unique index** : Ajouter des entrées si la valeur insérée n’existe pas ou mettre à jour les entrées si la valeur insérée existe déjà et qu’un risque de violation d’une clé unique se présente.

**Insert ignore** : Ajouter uniquement de nouvelles lignes afin d’empêcher les erreurs de doublons de clés.

**Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update et Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et cochez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

---

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Die on error

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Specify a data source alias

Cochez cette case et spécifiez l’alias de la source de données créée dans **Talend Runtime** pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans **Talend Runtime**. Pour voir un exemple de cas d’utilisation, consultez **Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647**.

**Avertissement :**

Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l’alias de la source de données spécifié.

Cette option est indisponible lorsque la case **Use an existing connection** est cochée.

### Advanced settings

#### Additional JDBC parameters

Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.

**Remarque :**

Vous pouvez appuyer sur **Ctrl+Espace** afin d’accéder à une liste de variables globales prédéfinies.

#### Extend Insert

Cochez cette case pour insérer un ensemble de lignes définies au lieu d’insérer les lignes une par une. Cette option permet un important gain de performance.
| **Number of rows per insert** | :saisissez le nombre de lignes à insérer en un bloc. Notez que si vous sélectionnez un nombre important de lignes, cela peut augmenter la quantité de mémoire utilisée et donc diminuer les performances.

*i Remarque :*

Cette option n'est pas compatible avec le lien *Reject*. Vous devez donc la décocher si vous utilisez un lien *Reject* en sortie du composant. |
|---|---|
| **Use Batch** | Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.

*i Remarque :*

Cette case est disponible uniquement si vous avez choisi l’option *Insert, Update, Single Insert Query* ou *Delete* dans le champ *Action on data*. |
| **Batch Size** | Spécifiez le nombre d’enregistrements à traiter dans chaque lot.

Ce champ est disponible uniquement lorsque la case *Use batch mode* est cochée. |
| **Commit every** | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution. |
| **Additional Columns** | Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier. |
| **Name** | Saisissez le nom de la colonne à modifier ou à insérer. |
| **SQL expression** | Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes. |
| **Position** | Sélectionnez *Before, Replace* ou *After*, en fonction de l’action à effectuer sur la colonne de référence. |
| **Reference column** | Saisissez une colonne de référence que le composant *tMysqlOutput* peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier. |
| **Use field options** | Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données. |
| **Use Hint Options** | Cochez cette case pour activer la zone de configuration des indicateurs (ou Hints) permettant d’optimiser |
l’exécution d’une requête. Dans cette zone, les paramètres sont :
- **HINT** : spécifiez l’indicateur dont vous avez besoin, en utilisant la syntaxe `/*+ */`.
- **POSITION** : spécifiez la place de l’indicateur dans une instruction SQL.
- **SQL STMT** : sélectionnez l’instruction SQL que vous souhaitez utiliser.

<table>
<thead>
<tr>
<th>Enable debug mode</th>
<th>Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.</th>
</tr>
</thead>
</table>
| Use duplicate key update mode insert | Met à jour les valeurs des colonnes spécifiées en cas de doublon de la clé primaire.  
**Column** : Saisissez entre guillemets le nom de la colonne à mettre à jour.  
**Value** : Saisissez l’opération que vous souhaitez effectuer sur la colonne.  
**Remarque** : Pour utiliser cette option, vous devez préalablement sélectionner le mode *Insert* dans la liste *Action on data* de la vue *Basic settings*. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.  
**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable *After* et retourne un entier.  
**NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable *After* et retourne un entier.  
**NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable *After* et retourne un entier.  
**NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable *After* et retourne un entier.  
**QUERY** : requête traitée. Cette variable est une variable *After* et retourne une chaîne de caractères.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

**Utilisation**

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Ce composant doit être utilisé en tant que composant de sortie. Il permet de faire des actions sur une table ou les données d’une table d’une base de données MySQL. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur. Pour un exemple d’utilisation du tMysqlOutput, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend. |

**Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput**

Ce scénario Java est un Job à trois composants permettant de créer de manière aléatoire des données de test à l’aide du composant tRowGenerator, dupliquer la colonne à modifier à l’aide du composant tMap et de modifier les données à insérer en utilisant une expression SQL dans le composant tMysqlOutput.
• Cliquez et déposez les composants suivants dans l’éditeur graphique : tRowGenerator, tMap et tMysqlOutput.

• Connectez-les à l’aide de liens Row Main.

• Dans l’éditeur graphique, double-cliquez sur le tRowGenerator pour afficher l’onglet Basic settings.

• Cliquez sur le bouton [...] à côté du champ Edit schema pour définir les données à transmettre au composant tMap, dans ce scénario, les deux colonnes name et random_date.

• Cliquez sur OK pour fermer la boîte de dialogue.

• Cliquez sur le bouton [...] à côté du champ RowGenerator Editor pour ouvrir l’éditeur du tRowGenerator pour paramétrer les données à générer.
Cliquez dans le champ Functions correspondant et sélectionnez une fonction pour chacune des deux colonnes, `getFirstName` pour la colonne `name` et `getRandomDate` pour la colonne `random_date`.

Dans le champ Number of Rows for Rowgenerator, saisissez 10 pour ne générer que dix lignes de prénoms et cliquez sur Ok pour fermer l'éditeur.

Double-cliquez sur le composant tMap pour ouvrir l'éditeur du tMap. L'éditeur s'ouvre et affiche les données d'entrée du composant tRowGenerator.

Dans le panneau Schema editor de l'éditeur du tMap, cliquez sur le bouton [+] de la table de sortie pour ajouter deux lignes et nommez la première colonne `random_date` et la deuxième `random_date1`. 
Dans ce scénario, l’objectif est de dupliquer la colonne `random_date` et adapter le schéma afin de modifier les données à transférer dans le composant de sortie.

- Dans l’éditeur du tMap, glissez la ligne `random_date` de la table d’entrée vers les lignes `random_date` et `random_date1` de la table de sortie.

- Cliquez sur OK pour fermer l’éditeur.

- Dans l’éditeur graphique, double-cliquez sur le composant tMysqlOutput pour afficher l’onglet Basic settings et paramétrer ses propriétés.
• Dans la liste Property Type, sélectionnez l’option Repository et cliquez sur le bouton [...] pour ouvrir la boîte de dialogue [Repository content] et sélectionnez la connexion à la base de données adéquate. Les informations de connexion sont renseignées automatiquement.

• Configurez les informations de connexion à la base de données dans les champs correspondants.

⚠️ Remarque :
Si vous n’avez pas stocké de connexion à votre base de données sous le nœud DB connections sous le nœud Metadata du Repository, sélectionnez l’option Built-in dans la liste Property Type et renseignez les informations de connexion manuellement.

• Cliquez sur le bouton [...] à côté du champ Table et sélectionnez la table à modifier : la table Dates dans ce scénario.

• Dans la liste Action on table, sélectionnez l’option Drop table if exists and create et sélectionnez Insert dans la liste Action on data.

• Si nécessaire, cliquez sur Sync columns pour récupérer le schéma de colonnes du composant tMap.

• Cliquez sur l’onglet Advanced settings pour afficher la vue correspondante et définir les paramètres avancés du composant.

• Dans la zone Additional Columns, paramétrez les modifications à appliquer aux colonnes.
Dans ce scénario, la colonne *One_month_later* remplace la colonne *random_date_1* et les données aussi sont modifiées à l’aide d’une requête SQL ajoutant un mois à la date générée aléatoirement dans la colonne *random_date_1*. Par exemple : 2007-08-12 devient 2007-09-12.

-Saisissez *One_Month_Later* dans la cellule *Name*.

-Dans la cellule *SQL expression*, saisissez la requête SQL ajoutant un mois, dans ce scénario :

```
"addate(Random_date, interval 1 month)"
```

-Sélectionnez *Replace* dans la liste *Position*.

-Sélectionnez *Random_date1* dans la liste *Reference column*.

**Remarque :**

Pour ce Job, vous dupliquez la colonne *random_date_1* dans la table avant de la remplacer par la colonne *One_Month_Later*. Le but de cette manipulation était de voir les modifications apportées en amont.

- Enregistrez le Job et appuyez sur **F6** pour l’exécuter.

La nouvelle colonne *One_month_later* remplace la colonne *random_date1* dans la base de données et ajoute un mois à chaque date générée aléatoirement.

<table>
<thead>
<tr>
<th>Random_date</th>
<th>One_Month_Later</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008-11-02 15:33:29</td>
<td>2008-12-02 15:33:29</td>
</tr>
<tr>
<td>2007-09-06 17:44:36</td>
<td>2007-10-06 17:44:36</td>
</tr>
<tr>
<td>2008-04-20 15:36:00</td>
<td>2008-05-20 15:36:00</td>
</tr>
</tbody>
</table>

**Scénario : Effectuer des mises à jour dans une base de données**

MySQL est utilisé à titre d’exemple.

Ce scénario est un Job à deux composants permettant de mettre à jour les données d’une table MySQL à partir d’un fichier délimité.

- A partir de la **Palette**, cliquez et déposez les composants *tFileInputDelimited* et *tMysqlOutput* dans l’éditeur graphique puis connectez-les à l’aide d’un lien *Row Main*. 
• Dans l’éditeur graphique, double-cliquez sur le **tFileInputDelimited** pour afficher l’onglet **Basic settings**.

• Dans le champ **Property Type**, sélectionnez le mode **Repository** si vous avez stocké les informations du fichier délimité sous le nœud **Metadata** du **Repository** ou sélectionnez le mode **Built-In** pour les définir manuellement. Dans ce scénario, utilisez le mode **Built-In**.

• Dans le champ **File Name**, renseignez manuellement le chemin d’accès au fichier contenant les mises à jour à propager dans la base de données ou cliquez sur le bouton […] pour parcourir vos dossiers jusqu’à ce fichier. Dans cet exemple, utilisez le fichier **customer_update**, qui comporte quatre colonnes, *id*, **CustomerName**, **CustomerAddress** et **idState**.
• Définissez si nécessaire les séparateurs de lignes (Row separator) et de champs (Field separator), l'en-tête (Header) et le pied-de-page (Footer), ainsi que le nombre de lignes à traiter (Limit). La première ligne, qui porte le nom des colonnes, est ignorée dans cet exemple, c'est pourquoi le champ Header est défini à "1".

• Cliquez sur le bouton [...] à côté du champ Edit Schema pour définir les données à passer au composant suivant. Dans cet exemple, le schéma est constitué de quatre colonnes, id, CustomerName, CustomerAddress et idState.

![Schema de données](image)

• Devant le nom de chaque colonne se trouve une case Key. Cochez cette case pour la ou les colonne(s) que vous voulez définir comme clé.

**Remarque :**

Vous devez obligatoirement définir au moins une clé pour que le Job s'exécute. Dans le cas contraire, le Job s'arrête automatiquement et un message d'erreur s'affiche dans la console de log.

• Dans l’éditeur graphique, double-cliquez sur le composant tMysqlOutput pour paramétrer ses propriétés dans l’onglet Basic settings de la vue Component :

![Paramètres de composant](image)

• Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton [...] à côté du champ Edit schema pour consulter le schéma et vérifier qu’au moins une colonne a été définie comme clé.

• Dans le champ Property Type, sélectionnez Repository si vous avez stocké les informations de connexion à la base de données sous le nœud Metadata du Repository ou sélectionnez le mode Built-In, puis renseignez manuellement les champs suivants : Host, Port, Database, Username et Password.
• Dans le champ **Table**, saisissez le nom de la table à mettre à jour.

• Dans le champ **Action on table**, sélectionnez l’opération que vous souhaitez effectuer sur la table. Pour ce scénario, sélectionnez **Default** car la table existe déjà.

• Dans le champ **Action on data**, sélectionnez l’opération que vous souhaitez effectuer sur les données. Pour ce scénario, sélectionnez l’option **Update** pour mettre la table à jour.

• Enregistrez le Job puis appuyez sur **F6** pour l’exécuter.

A partir de votre explorateur de base de données, vous pouvez vérifier que la table **customers** a bien été mise à jour. Comme avant sa mise à jour, elle se présente sous la forme des quatre colonnes id, **CustomerName**, **CustomerAddress** et **idState**, mais certains champs ont été modifiés selon les données du fichier délimité **customer_update**.

### Scénario : Récupérer les données erronées à l’aide d’un lien **Reject**

Ce scénario décrit un Job à quatre composants effectuant une migration d’un fichier client vers la table d’une base de données MySQL et redirigeant les données erronées vers un fichier CSV à l’aide d’un lien **Reject**.
Dans le Repository, sélectionnez la métadonnée correspondant au fichier client que vous souhaitez migrer et glissez-la dans l’espace de modélisation. Dans la boîte de dialogue [Components], sélectionnez tFileInputDelimited et cliquez sur OK. Les propriétés du composant seront automatiquement renseignées.

Si vous n’avez pas enregistré les informations concernant votre fichier client sous le nœud Metadata du Repository, glissez un composant tFileInputDelimited de la famille File > Input de la Palette et renseignez ses propriétés manuellement dans la vue Component.

A partir de la Palette, glissez un composant tMap de la famille Processing dans l’espace de modélisation.

Dans le Repository, développez le nœud Metadata et Db Connections, et sélectionnez la métadonnée de connexion à la base de données dans laquelle vous souhaitez migrer vos données et glissez-la dans l’espace de modélisation. Dans la boîte de dialogue [Components], sélectionnez tMysqlOutput et cliquez sur OK. Les propriétés de connexion à la base de données seront automatiquement renseignées.

Si vous n’avez pas enregistré les informations de connexion à votre base de données sous le nœud Db Connections du Repository, glissez un composant tMysqlOutput de la famille Databases de la Palette et renseignez ses propriétés manuellement dans la vue Component.

Pour plus d’informations, consultez le Guide utilisateur du Studio Talend :

A partir de la Palette, sélectionnez un composant tFileOutputDelimited de la famille File > Output et glissez-le dans l’espace de modélisation.

Reliez les composant customers et tMap, et les composants tMap et Localhost à l’aide d’un lien de type Row Main. Vous nommerez ce deuxième lien out.

Reliez le composant Localhost au tFileOutputDelimited à l’aide d’un lien de type Row > Reject.

Double-cliquez sur le composant customers pour afficher la vue Component correspondante.
Dans la liste Property Type, sélectionnez le mode Repository puis cliquez sur le bouton [...] correspondant pour sélectionner la métadonnée contenant la connexion à votre fichier. Sinon, sélectionnez le mode Built-in et paramétrez manuellement les champs suivants.

Cliquez sur le bouton [...] à côté du champ File Name, et indiquez le chemin d'accès et le nom du fichier à utiliser.

Dans les champs Row et Field Separator, saisissez entre guillemets les séparateurs de lignes et de champs utilisés dans le fichier.

Dans les champs Header, Footer et Limit, saisissez le nombre de ligne d'en-tête et de pied-de-page à ignorer, ainsi que le nombre de ligne limite à traiter.

Dans la liste Schema, sélectionnez Repository et cliquez sur le bouton [...] correspondant pour sélectionner le schéma de votre fichier, s'il est stocké sous le nœud Metadata du Repository. Vous pouvez aussi cliquer sur le bouton [...] correspondant au champ Edit schema et définir manuellement le schéma de votre fichier.

Le schéma se présente comme suit :
• Double-cliquez sur le tMap afin d'ouvrir son éditeur.

• Sélectionnez les colonnes id, CustomerName, CustomerAddress, idState, id2, RegTime, RegisterTime de la zone de gauche et glissez-les dans la table out de la zone de droite.
Dans la zone **Schema editor** en bas de l’éditeur du **tMap**, dans le tableau à droite, changez la longueur de la colonne **CustomerName** en 28 afin de créer une erreur. Ainsi, les données dont la longueur est supérieure à 28 créeront des erreurs qui pourront être récupérées grâce au lien **Reject**.

- Cliquez sur **OK**.

- Dans l’espace de modélisation, double-cliquez sur le composant de sortie **Localhost** pour afficher la vue **Component** correspondante.

Dans la liste **Property Type**, sélectionnez le mode **Repository** puis cliquez sur le bouton [...] correspondant pour sélectionner la métadonnée de connexion à la base de données. Les informations de connexion sont renseignées automatiquement. Sinon, sélectionnez le mode **Built-in** et paramétrez ces champs manuellement.

- Dans le champ **Table**, saisissez le nom de la table à créer. Dans ce scénario, vous l’appellez **customers_data**.

- Dans la liste **Action on data**, sélectionnez l’option **Create table**.

- Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
• Assurez-vous que la case **Die on error** est bien décochée, afin que le Job s’exécute malgré l’erreur que vous venez de créer.

• Cliquez sur l’onglet **Advanced settings** de la vue **Component** pour paramétrer les paramètres avancés du composant.

![Component settings](image)

• Décochez la case **Extend Insert** permettant d’insérer des lignes par bloc car cette option n’est pas compatible avec le lien **Reject**.

• Double-cliquez sur le composant **tFileOutputDelimited** pour paramétrer ses propriétés dans la vue **Component**.

![FileOutputDelimited settings](image)

• Cliquez sur le bouton [...] à côté du champ **File Name** pour indiquer le répertoire de destination et le nom du fichier de sortie.

• Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

• Enregistrez votre Job et appuyez sur **F6** pour l’exécuter.
Les données en erreur sont envoyées dans le fichier délimité, ainsi que le type d’erreur rencontrée. Ici, vous avez affaire à des données tronquées : **Data truncation.**
tMysqlOutputBulk

Ce composant écrit un fichier composé de colonnes et basé sur le séparateur défini et sur les standards MySQL.

Les composants tMysqlOutputBulk et tMysqlBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tMysqlOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Propriétés du tMysqlOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tMysqlOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tMysqlOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th><strong>Database</strong></th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>File Name</strong></td>
<td>Nom du fichier à générer.</td>
</tr>
<tr>
<td></td>
<td>Ce fichier est généré sur la même machine que le Studio Talend ou que le Job contenant un tMysqlOutputBulk.</td>
</tr>
<tr>
<td><strong>Append</strong></td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark,</td>
</tr>
</tbody>
</table>
évitez le mot réservé line lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

**Repository** : Le schéma existe déjà et il est stocké dans le *Repository*. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre *[Repository Content]*.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Row separator</strong></th>
<th>Chaîne (ex : &quot;\n&quot; sous Unix) séparant les lignes.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field separator</strong></td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td><strong>Text enclosure</strong></td>
<td>Caractères utilisés pour entourer le texte.</td>
</tr>
<tr>
<td><strong>Create directory if does not exist</strong></td>
<td>Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s’il n’existe pas déjà.</td>
</tr>
<tr>
<td><strong>Custom the flush buffer size</strong></td>
<td>Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ <em>Row number</em>, saisissez le nombre de lignes après lesquelles la mémoire est à nouveau libérée.</td>
</tr>
</tbody>
</table>
### Records contain NULL value
Cette case est cochée par défaut. Elle permet de prendre les champs de valeur NULL en compte. Si vous la décochez, les valeurs NULL seront remplacées par des valeurs vides.

### Check disk space
Cochez cette case afin de retourner une exception durant l’exécution si le disque est plein.

### Encoding
 Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec le composant tMysqlBulkExec. Ensemble, ils offrent un gain de performance important pour l’alimentation d’une base de données MySQL.</th>
</tr>
</thead>
</table>
Scénario : Insérer des données transformées dans une base MySQL

Ce scénario décrit un Job contenant quatre composants dont le but est d'alimenter un base MySQL à partir d'un fichier paramètres contenant des données transformées. Deux étapes sont requises pour ce Job, d'une part pour créer le fichier paramètres qui sera utilisé dans un second temps. La première étape inclut une phase de transformation des données contenues dans le fichier.

Déposer et relier les composants

Procédure
1. Glissez et déposez les composants suivants : \texttt{tRowGenerator}, \texttt{tMap}, \texttt{tMysqlOutputBulk} et \texttt{tMysqlBulkExec} dans l'espace de modélisation.
2. Connectez le flux principal à l'aide de connexions \texttt{Row > Main}.
3. Connectez le composant de début (\texttt{tRowGenerator}, dans cet exemple) au composant \texttt{tMysqlBulkExec} à l'aide d'une connexion \texttt{Trigger} de type \texttt{OnSubjobOk}.

Configurer les composants

Procédure
1. Un composant \texttt{tRowGenerator} est utilisé pour générer des données de façon aléatoire. Double-cliquez sur le composant \texttt{tRowGenerator} pour lancer l'éditeur.
2. Définissez le schéma des lignes à générer et la nature des données à générer. Dans cet exemple, le fichier \texttt{clients} à créer contient les colonnes suivantes : \texttt{ID}, \texttt{First Name}, \texttt{Last name}, \texttt{Address}, \texttt{City} qui sont toutes de type chaîne de caractères (\texttt{string}) à l'exception de l'\texttt{ID} qui est de type entier (\texttt{integer}).
Certaines informations du schéma n’ont pas nécessairement besoin d’être affichées. Pour les dissimuler, cliquez sur le bouton Columns dans la barre d’outils et décochez les colonnes à cacher, par exemple : Precision ou Parameters.

Utilisez le bouton [+] pour ajouter autant de colonnes que possible à votre schéma.

Cliquez sur le bouton Refresh en haut à droite de l’éditeur pour visualiser un aperçu des lignes générées en sortie dans l’onglet Preview en bas de l’éditeur.

3. Dans l’espace de modélisation, double-cliquez sur le composant tMap pour ouvrir son éditeur et paramétrer la transformation.

4. Faites glisser toutes les colonnes de la table d’entrée (row1) vers la table de sortie (clients).

5. Appliquez la transformation sur la colonne LastName en ajoutant .toUpperCase() à la fin. Cela mettra les noms des clients en majuscule.

Cliquez sur OK pour valider la transformation.

6. Dans l’espace de modélisation, double-cliquez sur le composant tMysqlOutputBulk pour afficher sa vue Component et paramétrer ses propriétés.

7. Définissez le chemin d’accès et le nom du fichier à produire dans le champ File Name. Si les propriétés du fichier délimité sont conservées dans le Repository, sélectionnez l’entrée correspondante dans le champ Property type afin de les récupérer. Dans cet exemple, le nom de fichier est clients.txt.
Le schéma est propagé à partir du composant **tMap**, si vous cliquez sur **Yes** lorsque la boîte de dialogue vous demande de confirmer la propagation. Sinon, cliquez sur le bouton **Sync columns** pour récupérer le schéma.

8. Dans cet exemple, n’incluez pas les informations d’en-tête, puisque la table doit déjà les contenir.

9. Cliquez sur **OK** pour valider la sortie.

10. Double-cliquez sur le composant **tMysqlBulkExec** pour paramétrer la requête INSERT à exécuter.


12. Dans le champ **Table**, saisissez le nom de la table à alimenter, ici, **clients**.

13. Dans l’onglet **Advanced settings**, saisissez le séparateur de colonnes, dans le champ **Fields terminated by**.

14. Vérifiez que l’encodage défini dans le champ **Encoding** correspond à celui de vos données.

### Sauvegarder et exécuter le Job

**Procédure**

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour exécuter le Job.

### Résultats

<table>
<thead>
<tr>
<th>ID</th>
<th>Firstname</th>
<th>Lastname</th>
<th>Address</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Theodore</td>
<td>EISENHOWER</td>
<td>Moreno Drive</td>
<td>Concord</td>
</tr>
<tr>
<td>2</td>
<td>Rutherford</td>
<td>COOLIDGE</td>
<td>Lindbergh Blvd</td>
<td>Madison</td>
</tr>
<tr>
<td>3</td>
<td>William</td>
<td>JEFFERSON</td>
<td>Katella Avenue</td>
<td>Boise</td>
</tr>
<tr>
<td>4</td>
<td>Benjamin</td>
<td>BUCHANAN</td>
<td>French Camp Turnpike Road</td>
<td>Juneau</td>
</tr>
<tr>
<td>5</td>
<td>Andrew</td>
<td>WILSON</td>
<td>North Preister Lane</td>
<td>Frankfort</td>
</tr>
<tr>
<td>6</td>
<td>Theodore</td>
<td>MCKINLEY</td>
<td>Pacific Hwy S</td>
<td>Augusta</td>
</tr>
<tr>
<td>7</td>
<td>William</td>
<td>POLK</td>
<td>Calle Feal</td>
<td>Tallahassee</td>
</tr>
<tr>
<td>8</td>
<td>Dwight</td>
<td>HAYES</td>
<td>Timberlane Drive</td>
<td>Des Moines</td>
</tr>
<tr>
<td>9</td>
<td>Abraham</td>
<td>BUCHANAN</td>
<td>Pache Boulevard</td>
<td>Des Moines</td>
</tr>
<tr>
<td>10</td>
<td>Abraham</td>
<td>MADISON</td>
<td>N Kentwood</td>
<td>Olympia</td>
</tr>
</tbody>
</table>
La table de la base clients est alimentée avec les données du fichier notamment les données transformées dans la colonne Last name.

Pour une simple opération d'Insert ne nécessitant pas de transformation, l'utilisation du composant tMysqlOutputBulkExec permet d'économiser une étape dans le processus et ainsi de gagner en performance.

Voir également Propriétés du tMysqlOutputBulkExec Standard à la page 2689.
tMysqlOutputBulkExec

Ce composant effectue une action d’Insert dans la base de données MySQL spécifiée.
En tant que composant dédié, le tMysqlBulkExec améliore les performances durant les opérations Insert dans une base de données MySQL.

Les composants tMysqlOutputBulk et tMysqlBulkExec sont généralement utilisés ensemble comme deux parties d’un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l’opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tMysqlOutputBulkExec.

Propriétés du tMysqlOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tMysqlOutputBulkExec s’exécutant dans le framework de Jobs Standard.
Le composant tMysqlOutputBulkExec Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version de MySQL que vous utilisez.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue.</td>
</tr>
</tbody>
</table>
qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :
| None : n’effectue aucune action sur la table. |
| Drop and create table : supprime la table puis en créer une nouvelle. |
| Create table : crée une table qui n’existe pas encore. |
| Create table if not exists : crée la table si nécessaire. |
| Drop table if exists and create : supprimer la table si elle existe déjà, puis en créer une nouvelle. |
| Clear table : supprime le contenu de la table. |

| Table | Nom de la table à écrire. |
| Remarque : Une seule table peut être écrite à la fois et cette table doit déjà exister pour que l’opération d’Insert soit autorisée. |

| Local FileName | Nom du fichier à générer et à traiter. |
| Ce fichier est généré sur la même machine que le Studio Talend ou que le Job contenant un tMysqlOutputBulkExec, puis chargé dans la base de données spécifiée dans le champ Host. |

| Append | Cochez cette option pour ajouter de nouvelles lignes à la fin du fichier. |

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository. |
| Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles : |
| View schema : sélectionnez cette option afin de voir le schéma. |
| Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales. |
| Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| **Built-In** | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |

### Advanced settings

| **Additional JDBC Parameters** | Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. |
| **Remarque :** | Vous pouvez appuyer sur Ctrl+Espace afin d’accéder à une liste de variables globales prédéfinies. |
| **Row separator** | Chaîne (ex : "\n" sous Unix) séparant les lignes. |
| **Field separator** | Caractère, chaîne ou expression régulière séparant les champs. |
| **Text enclosure** | Caractères entourant les données. |
| **Create directory if not exists** | Cette case est cochée par défaut. Cette option permet de créer le dossier contenant le fichier de sortie s’il n’existe pas déjà. |
| **Custom the flush buffer size** | Cochez cette case pour personnaliser la taille de la mémoire utilisée pour stocker temporairement les données, et dans le champ Row number, saisissez le nombre de lignes après lesquelles la mémoire est à nouveau libérée. |
| **Action on data** | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : |
| **Insert records in table** | Insère de nouveaux enregistrements dans une table. |
| **Replace records in table** | Remplace les enregistrements existants par de nouveaux. |
| **Update records in table** | Modifie les enregistrements existants. |
| **Ignore records in table** | Ignore les enregistrements existants ou insère les nouveaux. |
### Records contain NULL value
Cette case est cochée par défaut. Elle permet de prendre les champs de valeur NULL en compte. Si vous la décochez, les valeurs NULL seront remplacées par des valeurs vides.

### Encoding
 Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Utilisation

| Règle d’utilisation | Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans la base de données. |

### Scénario : Insérer des données en masse dans une base MySQL

Ce scénario est un Job à deux composants qui effectue le même type d’opération d’alimentation que les composants tMysqlOutputBulk (consultez tMysqlOutputBulk à la page 2682) et tMysqlBulkExec (consultez tMysqlBulkExec à la page 2602), mais sans transformation de données.

- Cliquez et déposez les composants suivants : tRowGenerator et tMysqlOutputBulkExec.
- Connectez les composants via un lien de type Row > Main.
- Paramétrez le tRowGenerator de la même manière que dans Scénario : Insérer des données transformées dans une base MySQL à la page 2685. Le schéma est constitué de plusieurs colonnes notamment : ID, First Name, Last Name, Address et City.
- A partir de l’espace de modélisation, double-cliquez sur le composant tMysqlOutputBulkExec pour afficher la vue Component et paramétrer ses propriétés.
Paramétrez la connexion à la base de données si nécessaire, en suivant les recommandations dans Scénario : Insérer des données transformées dans une base MySQL à la page 2685, concernant la conservation des informations de connexion dans la partie Metadata du Repository. Pour cela, sélectionnez Repository dans le champ Property Type et sélectionnez la connexion adéquate dans le champ adjacent. Les champs suivants seront renseignés automatiquement.

Pour plus d’informations, consultez le Guide utilisateur du Studio Talend.

Dans le champ Action on table, sélectionnez l’option None puisque vous souhaitez insérer les données dans une table déjà existante.

Dans le champ Table, saisissez le nom de la table à alimenter, ici, clients.

Dans le champ Local filename, indiquez le chemin d’accès et le nom du fichier contenant les données à charger dans la table, ici, clients.txt.

Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent.

Cliquez sur l’onglet Advanced settings pour paramétrer les propriétés avancées du composant.

Dans la liste Action on data, sélectionnez l’option Insert records in table pour insérer ces nouvelles données dans la table.

Appuyez sur F6 pour exécuter le Job

Le résultat devrait être relativement identique à celui obtenu dans Scénario : Insérer des données transformées dans une base MySQL à la page 2685. Cependant les données elles-mêmes peuvent différer légèrement puisque les données sont regénérées de façon aléatoire à chaque exécution.
tMysqlRollback

Ce composant annule le commit de transaction dans une base de données MySQL connectée pour éviter le commit de transaction involontaire.

Propriétés du tMysqlRollback Standard

Ces propriétés sont utilisées pour configurer le tMysqlRollback s'exécutant dans le framework de Jobs Standard.

Le composant tMysqlRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tMysqlConnection dans la liste s'il y a plus d'une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est généralement utilisé avec d'autres composants MySQL, notamment les composants tMysqlConnection et tMysqlCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez</td>
</tr>
</tbody>
</table>
dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

Lorsqu'un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.
tMysqlRow

Ce composant exécute la requête SQL sur la base de données MySQL spécifiée.

Selon la nature de la requête et de la base de données, le tMysqlRow agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Le tMysqlRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tMysqlRow Standard

Ces propriétés sont utilisées pour configurer le tMysqlRow s’exécutant dans le framework de Jobs Standard.

Le composant tMysqlRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version de MySQL que vous utilisez.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
</tbody>
</table>

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

**Repository** : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Nom de la table à traiter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query type</td>
<td>La requête peut être Built-in ou distante dans le Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder. <strong>Repository</strong> : Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.</td>
<td></td>
</tr>
<tr>
<td>Guess Query</td>
<td>Cliquez sur le bouton Guess Query pour générer la requête correspondant au schéma de votre table dans le champ Query.</td>
</tr>
<tr>
<td>Query</td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Rejects.</td>
</tr>
<tr>
<td>Specify a data source alias</td>
<td>Cochez cette case et spécifiez l’alias de la source de données créée dans Talend Runtime pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647. Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l’alias de la source de données spécifié. Cette option est indisponible lorsque la case Use an existing connection est cochée.</td>
</tr>
<tr>
<td><strong>Advanced settings</strong></td>
<td><strong>Additional JDBC parameters</strong></td>
</tr>
</tbody>
</table>
### Propagate QUERY’s recordset
Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste `use column`.

**Remarque:**
Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type `Object`. Ce composant est généralement suivi du `tParseRecordSet`.

### Use PreparedStatement

- **Parameter Index**: Saisissez la position du paramètre dans l’instruction SQL.
- **Parameter Type**: Saisissez le type du paramètre.
- **Parameter Value**: Saisissez la valeur du paramètre.

Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

### Commit every
Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

#### Global Variables

**`QUERY`**: requête traitée. Cette variable est une variable `Flow` et retourne une chaîne de caractères.

**`ERROR_MESSAGE`**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case `Die on error` est décochée, si le composant a cette option.

Une variable `Flow` fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Scénario : Supprimer et re-générer un index de table MySQL

Le scénario suivant crée un Job à quatre composants permettant de supprimer un index de table, effectuer un Insert dans cette table avant de re-générer l’index.

### Relier les composants

**Procédure**

1. Sélectionnez les composants suivants dans la **Palette** et déposez-le dans l’espace de modélisation : deux **tMysqlRow**, un **tRowGenerator** et un **tMysqlOutput**.
2. Connectez le premier composant **tMysqlRow** au **tRowGenerator** à l’aide d’un lien **OnComponentOk**.
3. Reliez le **tRowGenerator** au **tMysqlOutput** à l’aide d’un lien **Row > Main**.
4. Connectez le composant **tRowGenerator** au second composant **tMysqlRow** à l’aide d’une connexion **OnSubjobOk**.
Configurer les composants

Procédure

1. Sélectionnez le composant **tMysqlRow** et renseignez les propriétés de la base de données dans l’onglet **Basic settings**.
2. Dans les champs **Property type** et **Schema**, sélectionnez la base de données appropriée dans la liste.
   Les détails de connexion à la base de données ainsi que le schéma de la table sont renseignés automatiquement.
3. Propagez les informations de propriétés et de schéma aux autres composants du Job.
   La requête étant conservée dans les **Metadata** du Repository, vous pouvez également sélectionner **Repository** dans le champ **Query type** et cliquer sur la requête correspondante.
4. Si vous ne conservez pas vos requêtes dans le **Repository**, saisissez la déclaration SQL suivante : 
   ```sql
   drop index <index_name> on <table_name>
   ```
5. Sélectionnez le deuxième composant **tMysqlRow**, vérifiez ses propriétés et son schéma.
6. Puis saisissez la déclaration SQL permettant de recréer un index à l’aide de la formulation suivante :
   ```sql
   create index <index_name> on <table_name> (<column_name>);
   ```
   Le composant **tRowGenerator** permet de générer automatiquement les colonnes à ajouter à la table de sortie définie.
7. Sélectionnez le composant **tMysqlOutput** et renseignez les propriétés de connexion à la base de données soit à partir du Repository ou manuellement si les informations de connexion ne sont utilisées que pour ce Job. La table à alimenter a pour nom : **comprehensive**.
8. Le schéma est automatiquement hérité du flux de données du **tLogRow**. Editez le schéma et vérifiez que sa structure correspond au schéma attendu par la table de base de données spécifiée.
   Dans le champ **Action on table**, sélectionnez **None** et dans le champ **Action on data**, sélectionnez **Insert**.
   Aucune colonne supplémentaire n’est requise pour ce Job.

Exécuter le Job

Procédure

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter ce Job.

   Si vous avez la possibilité d’observer l’action sur la base de données, remarquez que l’index a été supprimé en début de Job puis recréé à la fin de l’action d’Insert.

   Voir également **tDBSQLRow** à la page 659.

**Scénario 2 : Utiliser l’instance PreparedStatement pour faire une requête sur des données**

Le scénario suivant décrit un Job à quatre composants permettant de lier la colonne d’une table à un fichier clients. En effet, la table MySQL contient la liste de tous les États américains avec leur identifiant, et le fichier contient des informations sur vos clients avec l’identifiant de l’État dans lequel ils résident. L’objectif de ce scénario est de récupérer le nom de l’État pour chaque client à l’aide d’une requête SQL. Afin de traiter un grand nombre de données plus rapidement, utilisez l’objet PreparedStatement permettant de n’exécuter qu’une seule fois la requête au lieu de l’effectuer à chaque ligne. Chaque ligne de données est envoyée comme paramètre. Un objet PreparedStatement peut également être utilisé pour éviter une injection SQL.

Pour ce scénario, utilisez un fichier et une base de données dont vous avez préalablement stocké la connexion et les propriétés dans des métadonnées dans le **Repository**. Pour plus d’informations concernant la création de métadonnées de fichiers délimités, la création de métadonnées de connexion à une base de données et l’utilisation de ces métadonnées, consultez le **Guide utilisateur du Studio Talend**.

**Relier les composants**

**Procédure**

1. A partir du **Repository**, développez le nœud **Metadata** et **File delimited**.
2. Sélectionnez la métadonnée correspondant au fichier clients et glissez cette métadonnée dans l’espace de modélisation. Ici, vous utilisez la métadonnée **customers**.
3. Double-cliquez sur **tFileInputDelimited** dans la boîte de dialogue [Components] afin de créer un composant **tFileInputDelimited** déjà paramétré.
4. Glissez **tMysqlRow**, **tParseRecordSet** et **tFileOutputDelimited** dans l’espace de modélisation.
5. Reliez **tFileInputDelimited** à **tMysqlRow** via un lien de type **Row > Main**.
6. Reliez **tMysqlRow** à **tParseRecordSet** via un lien de type **Row > Main**.
7. Reliez **tParseRecordSet** à **tFileOutputDelimited** via un lien de type **Row > Main**.

![](https://via.placeholder.com/150)

**Configurer les composants**

**Procédure**

1. Double-cliquez sur **tFileInputDelimited** pour ouvrir la vue **Basic settings**.
Dans la liste **Schema**, sélectionnez **Built-in** afin de modifier le schéma de votre composant. Puis, cliquez sur le bouton [...] à côté du champ **Edit schema** pour ajouter la colonne dans laquelle insérer le nom de l’État.

Cliquez sur le bouton [+] pour ajouter une colonne au schéma. Renommez cette colonne **LabelStateRecordSet**, et elle est de type objet, donc sélectionnez **Object** dans la liste **Type**. Cliquez sur **OK** pour enregistrer vos modifications.

A partir de la **Palette**, sélectionnez les composants **tMysqlRow**, **tParseRecordSet** et **tFileOutputDelimited** et glissez-les dans l’espace de modélisation.

Double-cliquez sur le composant **tMysqlRow** pour paramétrer ses propriétés dans l’onglet **Basic settings** de la vue **Component**.

7. Dans la liste Schema, sélectionnez Built-in pour paramétrer le schéma manuellement et ajouter la colonne LabelStateRecordSet, ou cliquez directement sur le bouton Sync columns pour récupérer le schéma du composant précédent.

8. Dans le champ Query, saisissez la requête SQL à utiliser. Ici, vous souhaitez récupérer les noms des États américains contenus dans la colonne LabelState de la table MySQL us_state : "SELECT LabelState FROM us_state WHERE idState=?". Le point d'interrogation "?" représente le paramètre à définir dans l'onglet Advanced settings.

10. Cochez la case **Propagate QUERY's recordset** et sélectionnez la colonne *LabelStateRecordSet* dans la liste **use column** pour insérer le résultat de la requête dans cette colonne.

11. Cochez la case **Use PreparedStatement** et définissez le paramètre utilisé dans la requête dans le tableau **Set PreparedStatement Parameters**.

12. Cliquez sur le bouton [+] pour ajouter un paramètre.
   a) IDans la cellule **Parameter Index**, saisissez la position du paramètre dans l'instruction SQL. Saisissez "1" étant donné que vous n'utilisez qu'un seul paramètre dans l'exemple.
   b) Dans la cellule **Parameter Type**, saisissez le type du paramètre. Ici, le paramètre est de type entier, donc sélectionnez **Int** dans la liste.
   c) Dans la cellule **Parameter Value**, saisissez la valeur du paramètre. Ici, vous souhaitez récupérer le nom de l'Etat en fonction de leur ID pour chaque client du fichier d'entrée, saisissez donc " row1.idState ".

13. Double-cliquez sur le composant **tParseRecordSet** pour paramétrer ses propriétés dans l'onglet **Basic settings** de la vue **Component**.

   b) Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent et le tableau **Attribute table** est automatiquement renseigné avec les colonnes du schéma.
   c) Dans le tableau **Attribute table**, dans le champ **Value** correspondant à la colonne *LabelStateRecordSet*, saisissez entre guillemets le nom de la colonne contenant le nom des Etats à récupérer et à mettre en correspondance avec chaque client. Dans cet exemple, saisissez "*LabelState*".

14. Double-cliquez sur le composant **tFileOutputDelimited** pour paramétrer ses propriétés dans l'onglet **Basic settings** de la vue **Component**.
a) Dans le champ **File Name**, renseignez le chemin d'accès et le nom du fichier de sortie.

b) Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour enregistrer le Job.
2. Appuyez sur **F6** pour l'exécuter.

**Résultats**

Une colonne contenant le nom de l’Etat américain correspondant à chaque client a été ajoutée au fichier.

**Scénario : Combiner deux flux pour une sortie sélective**

Dans ce scénario, un flux généré par un **tFixedFlowInput** est combiné avec un flux de la base de données MySQL. Le flux source contient les champs **id** et **age** alors que la table MySQL contient **id** et **name**. Vous allez récupérer les données **age** du flux source et les combiner avec les enregistrements **id** et **name** de la table MySQL à partir de la mise en correspondance avec **id**. Le schéma d’entrée est différent de celui de sortie, dans le **tMysqlRow**.
**Relier les composants**

**Procédure**

1. Déposez un `tFixedFlowInput`, un `tMysqlRow`, un `tParseRecordSet` et un `tLogRow` de la Palette dans l'espace de modélisation graphique.
2. Renommez le `tFixedFlowInput` en `source_flow`, le `tMysqlRow` en `insert_recordset`, le `tParseRecordSet` en `parse_recordset` et le `tLogRow` en `show_combined_flow`.
3. Reliez le `tFixedFlowInput` au `tMysqlRow` à l'aide d'un lien `Row > Main`.
4. Reliez le `tMysqlRow` au `tParseRecordSet` à l'aide d'un lien `Row > Main`.
5. Connectez le `tParseRecordSet` au `tLogRow` à l'aide d'un lien `Row > Main`.

![Diagramme de modélisation graphique](image.png)

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le `tFixedFlowInput` pour ouvrir sa vue `Basic settings`.

   ![Vue Basic settings du `tFixedFlowInput`](image.png)

   - **Mode**
     - Use Single Table
     - Use Inline Table
     - Use Inline Content (delimited file)
   - **Row Separator** : 
     - Use \n
   2. Sélectionnez l'option Use Inline Content (delimited file) dans la zone Mode. Dans le champ Content, saisissez les données à transférer :

   ```
   1;30
   2;20
   ```

Cliquez sur le bouton [+] pour ajouter deux colonnes, nommées id et age, de type Integer.
Cliquez sur OK pour fermer l’éditeur.

4. Double-cliquez sur le tMysqlRow pour ouvrir sa vue Basic settings.

5. Dans les champs Host et Port, saisissez les informations de connexion.
   Dans le champ Database, saisissez le nom de la base de données.
   Dans les champs Username et Password, saisissez les informations d’authentification.
   Dans le champ Query, saisissez la requête SQL permettant de récupérer les données de id et name dans la table MySQL employee : "select id, name from employee WHERE id=?".
   Le point d’interrogation, "?" représente le paramètre à configurer l’onglet Advanced settings.

7. Cliquez sur le bouton [+] pour ajouter deux colonnes, à droite, nommées recordset et age, de type Object et Integer. Notez que recordset doit contenir les résultats de la requête de la table MySQL, c'est-à-dire les champs id et name.
Cliquez sur OK pour fermer l'éditeur.

8. Cliquez sur l'onglet Advanced settings afin de configurer les paramètres avancés.

9. Cochez la case Propagate QUERY's recordset et sélectionnez recordset dans la liste use column afin d'insérer les résultats de la requête dans cette colonne.
Cochez la case Use PreparedStatement et définissez les paramètres utilisés dans la requête, dans la table Set PreparedStatement Parameters.

10. Cliquez sur le bouton [+] pour ajouter une ligne.
Dans la cellule Parameter Index, saisissez la position du paramètre dans l'instruction SQL.
Saisissez “1” puisque vous utilisez un paramètre dans cet exemple.
Dans la cellule Parameter Type, saisissez le type de paramètre. Le paramètre est de type Integer.
Sélectionnez Int dans la liste.
Dans la cellule Parameter Value, saisissez la valeur du paramètre. Ici, vous allez récupérer les colonnes id et name de la table employee selon la valeur de id du flux source. Saisissez row3.id.

11. Double-cliquez sur le composant tParseRecordSet pour ouvrir sa vue Basic settings.
Dans la liste **Prev. Comp. Column list**, sélectionnez la colonne à analyser, *recordset*.

12. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

Cliquez trois fois sur le bouton [+] pour ajouter trois colonnes, à droite, nommées *id*, *name* et *age*, respectivement de type *Integer*, *String* et *Integer*. Les colonnes *id* et *name* doivent contenir les données analysées de *recordset*.

Cliquez sur **OK** pour fermer l'éditeur.

Dans la table **Attribute table**, dans les champs **Value** correspondants aux colonnes *id* et *name*, saisissez le nom des colonnes de la table MySQL à récupérer, “id” et “name”.

13. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage sous forme de tableau.
Exécuter le Job

Procédure

1. Appuyez sur les touches Ctrl+S pour sauvegarder votre Job.
2. Appuyez sur F6 pour exécuter le Job.

```
[statistics] connecting to socket on port 4030
[statistics] connected
+----------+
<table>
<thead>
<tr>
<th>show_combined_flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>
+----------+
```

[statistics] disconnected
tMysqlSCD

Ce composant reflète et traque les modifications d’une table SCD MySQL dédiée.
Le tMysqlSCD répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.

**Propriétés du tMysqlSCD Standard**

Ces propriétés sont utilisées pour configurer le tMysqlSCD s’exécutant dans le framework de Jobs Standard.
Le composant tMysqlSCD Standard appartient aux familles Business Intelligence et Databases.
Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Sélectionnez la version de Mysql que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d'authentification de l'utilisateur de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td><strong>None</strong>: n'effectuer aucune opération de table.</td>
</tr>
<tr>
<td></td>
<td><strong>Create a table</strong>: créer une table qui n'existe pas encore.</td>
</tr>
<tr>
<td></td>
<td><strong>Create table if doesn't exist</strong>: créer la table si nécessaire.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong>: sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong>: sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong>: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
</tbody>
</table>
**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


**SCD Editor**
L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension.
Pour plus d’informations, consultez Méthodologie de gestion du SCD à la page 2716.

**Use memory saving mode**
Cochez cette case pour améliorer les performances du système.

**Source keys include Null**
Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null.

*Avertissement :
Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.*

**Die on error**
Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

**Advanced settings**

**Additional JDBC Parameters**
Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n’est pas disponible lorsque vous utilisez l’option Use an existing connection dans les Basic settings.

**End date time details**
Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format HH:mm:ss. La valeur par défaut pour ce champ est 12:00:00.

Ce champ apparaît uniquement lorsqu’un SCD de Type 2 est utilisé et lorsque Fixed year value est sélectionné pour créer la date de fin du SCD.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

**Debug mode**
Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.

**Variables globales**

**Global Variables**

- **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.
- **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.
**NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable *After* et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

---

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type <em>Row &gt; Main</em> en entrée.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton [*+] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. 

La table *Dynamic settings* est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue *Basic settings*. Lorsqu’un paramètre dynamique est configuré, la liste *Component List* de la vue *Basic settings* devient inutilisable. 

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez *Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte* à la page 2641 et *Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement* à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*. |
| Limitation          | Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD. |
Méthodologie de gestion du SCD

Lors de dimensions à évolution lente ou Slowly Changing Dimensions (SCD), les données changent lentement.

L'éditeur SCD permet de construire de manière simple le flux de sortie des données SCD. Dans l'éditeur SCD, vous pouvez relier les colonnes, sélectionner la clé de substitution (Surrogate key) et paramétrer les attributs des colonnes modifiées en combinant les différents types de modifications SCD. La figure ci-dessous représente l'éditeur SCD.

Clés SCD

Vous devez sélectionner une ou plusieurs colonnes qui serviront de clés permettant d'assurer l'unicité des données entrantes.

Vous devez aussi sélectionner une colonne sur laquelle positionner une clé de substitution (surrogate key) dans la table SCD et la relier à une des colonnes d’entrée de la table source. La valeur de la clé de substitution permet de relier les enregistrements de la table source aux enregistrements de la table SCD. L'éditeur utilise ce mapping pour localiser l’enregistrement dans la table SCD et pour déterminer si un enregistrement est nouveau ou s’il a été modifié. La clé de substitution est
généralement la clé primaire de la table source, mais elle peut aussi correspondre à une autre clé à partir du moment où elle permet d’identifier de façon unique un enregistrement et où ses valeurs ne changent pas.

**Source keys** : Glissez une ou plusieurs colonnes du panneau **Unused** vers le panneau **Source keys**. Ces colonnes seront utilisées en tant que clé(s) assurant l’unicité des données entrantes.

**Surrogate keys** : Paramétrez la colonne dans laquelle la clé de substitution générée sera stockée. Une clé de substitution peut être générée en fonction de la méthode sélectionnée dans la liste **Creation**.

**Creation** : Sélectionnez une des méthodes suivantes permettant de générer la clé.

- **Auto increment** : la clé est auto-incrémentée.
- **Input field** : la clé est fournie par un champ d’entrée.
  
  Lorsque cette méthode est sélectionnée, vous pouvez glisser le champ correspondant du panneau **Unused** vers le champ **complement**.

- **Routine** : à partir du champ **complement**, vous pouvez appuyer sur **Ctrl+Espace** afin d’afficher la liste d’auto-complétion et de sélectionner la routine appropriée.

- **Table max +1** : la valeur maximum de la table SCD est incrémentée pour créer une clé de substitution.

- **DB Sequence** : à partir du champ **complement**, vous pouvez saisir le nom de la séquence de la base de données (**DB Sequence**) existante qui va incrémenter automatiquement la colonne indiquée dans le champ **name**.

  L’option **DB Sequence** est disponible uniquement depuis l’éditeur **SCD** du composant **tOracleSCD**.

**Types de SCD**

Il existe quatre types de dimensions à évolution lente (Slowly Changing Dimensions) : du **Type 0** au **Type 3**. Vous pouvez, d’un simple glisser-déposer, appliquer n’importe quel type de SCD à n’importe quelle colonne de la table source.

- **Type 0** : ce type de SCD n’est pas beaucoup utilisé. Certaines données dimensionnelles peuvent être écrasées et d’autres peuvent rester inchangées au cours du temps. Ce type de SCD convient lorsque aucun effort n’a été mis en place pour gérer les dimensions à caractère évolutif.

- **Type 1** : no history is kept in the database. New data overwrites old data. Use this type if tracking changes is not necessary. this is most appropriate when correcting certain typos, for example the spelling of a name.

- **Type 2** : l’intégralité de l’historique est stockée dans la base de données. Ce type de SCD traque les données d’historique en enregistrant un nouvel enregistrement dans la table de dimension avec une nouvelle clé à chaque fois qu’un changement est effectué. Ce type de SCD convient lorsque l’on traque les mises à jour, par exemple.

- **Type 3** : seules les informations sur l’ancienne valeur d’une dimension est écrite dans la base de données. Ce type de SCD traque les changements en utilisant des colonnes séparées. Ce type de SCD convient lorsque l’on traque les valeurs précédentes d’une colonne qui change.

Le principe du SCD **Type 2** réside dans le fait qu’un nouvel enregistrement est ajouté à la table SCD lorsqu’un changement est détecté dans les colonnes sélectionnées. Notez que bien que plusieurs changements peuvent être effectués au même enregistrement sur plusieurs colonnes paramétrées en **Type 2**, une seule ligne traquant ces changements est ajoutée à la table SCD.
Le schéma du SCD Type 2 devrait inclure des colonnes spécifiques au SCD contenant les informations de log standard, notamment :

- **start** : ajoute une colonne au schéma de la table SCD contenant la date de début d’un enregistrement. Vous pouvez sélectionner une des colonnes du schéma d’entrée comme date de départ pour la table SCD.

- **end** : ajoute une colonne au schéma de la table SCD contenant la date de fin d’un enregistrement. Lorsque l’enregistrement est en cours, la date de fin est de valeur NULL, sinon vous pouvez utiliser une année fixe en sélectionnant Fixed Year Value dans la liste et renseignez l’année fictive dans la cellule d’à côté pour éviter d’avoir une valeur nulle dans le champ end.

- **version** : ajoute une colonne au schéma de la table SCD contenant le numéro de version de l’enregistrement.

- **active** : ajoute une colonne au schéma de la table SCD contenant les statuts true ou false. Cette colonne permet de repérer facilement les enregistrements actifs.

**Scénario : Identifier des modifications de données en utilisant les dimensions à évolution lente (SCD) de type 0 à 3**

Ce scénario décrit un Job qui stocke et gère à la fois les données actuelles et les données historiques des employés dans une table MySQL en utilisant les dimensions à évolution lente (Slowly Changing Dimensions).

Les données entrantes contiennent plusieurs informations sur les employés, comme leur nom (name), leur âge (age), leur rôle (role) et leur salaire (salary). Une colonne id est ajoutée afin d’assurer l’unicité des données entrantes.

Les données suivantes des employés sont d’abord insérées dans une nouvelle table MySQL en utilisant les dimensions à évolution lente :

```
id;name;age;role;salary
1;Mark Smith;30;tester;11000.00
2;Thomas Johnson;32;developer;12000.00
3;Teddy Brown;33;tester;13000.00
```

La table est ensuite mise à jour en utilisant les dimensions à évolution lente avec les données renouvelées suivantes des employés.

```
id;name;age;role;salary
1;Mark Smith;31;tester;11000.00
2;Thomas Johnson;32;developer;12000.00
3;Teddy Brown;33;writer;13500.00
```

Vous pouvez constater que l’âge de Mark Smith est mis à jour et passe de 30 à 31, que le rôle de Teddy Brown change de tester à writer et que son salaire de 13000.00 augmente à 13500.00. Dans ce scénario,

- vous ne souhaitez pas identifier les modifications des données pour le champ name (il faut donc exécuter le SCD de type 0 sur ces données) ;

- vous souhaitez que les nouvelles données age écrasent les données existantes (il faut donc exécuter le SCD de type 1 sur ces données) ;
• vous souhaitez conserver un historique exhaustif des données role, créer systématiquement un nouvel enregistrement avec les données modifiées et fermer les enregistrements précédents (il faut donc exécuter le SCD de type 2 sur ces données) et,

• vous souhaitez garder les précédentes valeurs de dimension pour le champ salary (il faut donc exécuter le SCD de type 3 sur ces données).

Pour plus d’informations concernant les types de SCD, consultez Méthodologie de gestion du SCD à la page 2716.

**Créer un Job pour identifier les modifications de données dans MySQL en utilisant les dimensions à évolution lente (SCD)**

Créez un Job pour ouvrir une connexion à une base de données MySQL, insérer les données des employés dans une table de la base de données MySQL en utilisant les dimensions à évolution lente.

Ce Job récupère et affiche ensuite les données saisies dans la console, met à jour les données des employés dans MySQL en utilisant les dimensions à évolution lente, récupère et affiche les données mises à jour dans la console et ferme la connexion à la base de données MySQL.
**Procédure**

1. Créez un nouveau Job et ajoutez un composant `tMysqlConnection`, deux composants `tFixedFlowInput`, `tMysqlSCD`, `tMysqlInput`, `tLogRow` et un composant `tMysqlClose` en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.

2. Reliez le premier composant `tFixedFlowInput` au premier `tMysqlSCD` via une connexion `Row > Main`.

3. De la même manière, reliez le premier composant `tMysqlInput` au premier `tLogRow`, le second `tFixedFlowInput` au second `tMysqlSCD`, ainsi que le second `tMysqlInput` au second `tLogRow`.

4. Reliez le premier `tMysqlConnection` au premier composant `tFixedFlowInput` à l’aide d’un lien `Trigger > OnSubjobOk`.

5. De la même manière, reliez le premier composant `tFixedFlowInput` au premier `tMysqlInput`, le premier composant `tMysqlInput` au second `tFixedFlowInput`, le second composant `tFixedFlowInput` au second `tMysqlInput` et le second composant `tMysqlInput` au second `tMysqlClose`.

**Ouvrir une connexion à la base de données MySQL**

Configurez le composant `tMysqlConnection` pour ouvrir une connexion à la base de données MySQL.

**Procédure**

1. Double-cliquez sur le composant `tMysqlConnection` pour ouvrir sa vue `Basic settings`.

2. Dans les champs `Host`, `Port`, `Database`, `Username` et `Password`, saisissez les informations requises pour la connexion à la base de données MySQL.

**Saisir les données des employés dans une table MySQL en utilisant les dimensions à évolution lente (SCD)**

Configurez le premier composant `tFixedFlowInput` et le premier `tMysqlSCD` afin d’insérer les données des employés dans la table de la base de données MySQL en utilisant les dimensions à évolution lente.

**Procédure**

1. Double-cliquez sur le premier composant `tFixedFlowInput` pour ouvrir sa vue `Basic settings`. 
2. Cliquez sur le bouton à côté de Edit schema et, dans la fenêtre qui s’ouvre, définissez le schéma en ajoutant 5 colonnes : id et age de type Integer, name et role de type String et salary de type Double.

Cela fait, cliquez sur OK pour sauvegarder les modifications. Dans la boîte de dialogue, cliquez sur Yes pour propager le schéma au composant suivant.

3. Dans la zone Mode, sélectionnez Use Inline Content (delimited file). Saisissez ensuite les données d’entrée suivantes des employés dans le champ Content qui s’affiche.

```
1;Mark Smith;30;tester;11000.00
2;Thomas Johnson;32;developer;12000.00
3;Teddy Brown;33;tester;13000.00
```

4. Cliquez sur le premier composant tMysqlSCD afin d’ouvrir sa vue Basic settings.
5. Cochez la case **Use an existing connection** et, dans la liste déroulante **Component List** qui s’affiche, sélectionnez le composant de connexion que vous avez configuré.

6. Dans le champ **Table**, saisissez `employee_scd`.

7. Cliquez sur le bouton à côté de **SCD Editor** pour ouvrir l’éditeur SCD. Toutes les colonnes du schéma sont listées dans le panneau **Unused**.

8. Dans le champ **name** du panneau **Surrogate keys**, saisissez le nom des clés de substitution, `SK` dans cet exemple.

9. Dans le panneau **Unused**, glissez-déposez :
   - `id` vers le panneau **Source keys** afin de l’utiliser comme une clé assurant l’unicité des données entrantes ;
   - `name` vers le panneau **Type 0 fields** (aucune action particulière ne sera exécutée sur ses changements de dimension) ;
   - `age` vers le panneau **Type 1 fields** pour exécuter un SCD Type 1 ;
   - `role` vers le panneau **Type 2 fields** pour exécuter un SCD Type 2 et
   - `salary` vers le panneau **Type 3 fields** pour exécuter un SCD Type 3.
Dans le panneau **Versioning**, cochez la case **version** pour renseigner les numéros de version de s enregistrements anciens et actuels dans la table SCD. Cochez également la case **active** pour ajouter la colonne qui renseigne la valeur **True** de l’enregistrement actif actuel ou la valeur **False** pour les anciens enregistrements dans la table SCD.

Cela fait, cliquez sur **OK** pour sauvegarder vos modifications et fermer l’éditeur SCD.

**Récupérer de MySQL les données insérées des employés**

Configurez le premier composant **tMysqlInput** et le premier composant **tLogRow** pour récupérer les données insérées des employés de la table SCD MySQL et les afficher dans la console.

**Procédure**

1. Double-cliquez sur le composant **tMysqlInput** afin d’ouvrir sa vue **Basic settings**.
2. Cochez la case **Use an existing connection** et, dans la liste déroulante **Component List** qui s’affiche, sélectionnez le composant de connexion que vous avez configuré.

3. Dans le champ **Table Name**, saisissez le nom de la table depuis laquelle les données des employés sont récupérées. Dans cet exemple, la table `employee_scd`.

4. Ouvrez l’éditeur de schéma du composant **tMysqlSCD**, sélectionnez et copiez toutes ses colonnes, ouvrez ensuite l’éditeur de schéma du composant **tMysqlInput** et définissez son schéma en collant les colonnes que vous avez copiées du composant **tMysqlSCD**.

Cela fait, cliquez sur **OK** pour sauvegarder les modifications. Dans la boîte de dialogue qui s’ouvre, cliquez sur **Yes** pour propager le schéma au composant suivant.

5. Cliquez sur le bouton **Guess Query** pour renseigner le champ **Query** à l’aide de la requête SQL générée automatiquement extrayant les données de toutes les colonnes de la table spécifiée.
6. Double-cliquez sur le premier composant tLogRow afin d’ouvrir sa vue Basic settings. Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour lire plus facilement les résultats affichés dans la console.

**Mettre à jour les données des employés dans MySQL en utilisant les dimensions à évolution lente (SCD)**

Configurez le second composant tFixedFlowInput et le second composant tMysqlSCD afin de mettre à jour les données des employés dans MySQL, en utilisant les dimensions à évolution lente.

**Procédure**

1. Double-cliquez sur le second composant tFixedFlowInput pour ouvrir sa vue Basic settings.
2. Cliquez sur le bouton à côté de Edit schema et, dans la fenêtre qui s’ouvre, définissez le schéma en copiant-collant celui du premier composant tFixedFlowInput.
3. Dans la zone Mode, sélectionnez Use Inline Content (delimited file). Saisissez ensuite les données mises à jour suivantes des employés dans le champ Content qui s’affiche.

```
1;Mark Smith;31;tester;11000.00
2;Thomas Johnson;32;developer;12000.00
3;Teddy Brown;33;writer;13500.00
```

4. Répétez l’4 à la page 2721 jusqu’à l’10 à la page 2723 dans la Saisir les données des employés dans une table MySQL en utilisant les dimensions à évolution lente (SCD) à la page 2720, afin de configurer le second composant tMysqlSCD.

**Récupérer les données mises à jour des employés dans MySQL**

Configurez le second composant tMysqlInput et le second composant tLogRow pour récupérer les données mises à jour des employés dans la table SCD MySQL et les afficher dans la console.

**Procédure**

1. Répétez 1 à la page 2723 jusqu’à 5 à la page 2724 dans la procédure Récupérer de MySQL les données insérées des employés à la page 2723 afin de configurer le second composant tMysqlInput.
2. Répétez 6 à la page 2725 dans la procédure Récupérer de MySQL les données insérées des employés à la page 2723, afin de configurer le second composant tLogRow.

**Fermeture de la connexion à une base de données MySQL**

Configurez le composant tMysqlClose pour fermer la connexion à une base de données MySQL.

**Procédure**

1. Double-cliquez sur le composant tMysqlClose afin d’ouvrir sa vue Basic settings.
2. Dans la liste déroulante Component List, sélectionnez le composant de connexion qui ouvre la connexion que vous souhaitez fermer. Dans cet exemple, le composant tMysqlConnection_1.
Exécutez le Job pour identifier les modifications de données dans MySQL en utilisant les dimensions à évolution lente (SCD)

Une fois le Job et les composants utilisés dans le Job pour identifier les modifications de données dans MySQL en utilisant les dimensions à évolution lente configurés, vous pouvez exécuter le Job et vérifier les résultats de l'exécution du Job.

Procédure

1. Appuyez sur Ctrl + S pour sauvegarder le Job.
2. Appuyez sur F6 pour exécuter le Job.

Comme affiché ci-dessus, la nouvelle donnée age, 31, pour Mark Smith écrase simplement l'ancienne donnée age, 30. Un nouvel enregistrement avec une valeur SK définie à 4 est créé pour la modification du role pour Teddy Brown qui change de tester à writer. Enfin, les valeurs du salaire antérieur et du salaire actuel de Teddy Brown sont toutes les deux inscrites dans le nouvel enregistrement.
Ce composant reflète et traque les modifications d’une table MySQL SCD dédiée.

Le tMysqlSCDELT répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table MySQL SCD dédiée.

**Propriétés du tMysqlSCDELT Standard**

Ces propriétés sont utilisées pour configurer le tMysqlSCDELT s’exécutant dans le framework de Jobs Standard.

Le composant tMysqlSCDELT Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>DB Version</strong></td>
<td>Sélectionnez la version de Mysql que vous utilisez.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Source table</td>
<td>Nom de la table contenant les données à filtrer.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois pour que l'opération d'insert soit autorisée.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :

- **None** : n'effectuer aucune opération de table.
- **Drop and create the table** : supprimer la table puis en créer une nouvelle.
- **Create a table** : créer une table qui n'existe pas encore.
- **Create table if doesn't exist** : créer la table si nécessaire.
- **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.
- **Clear a table** : supprimer le contenu de la table.
- **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Built-in
Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository

#### Source Keys
Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l’unicité des données entrantes.

#### Use SCD Type 1 fields
Utilisez le type 1 si vous n’avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.

#### Use SCD Type 2 fields
Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.

- **Start date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l’une des colonnes d’entrée du schéma comme date de départ (Start Date) dans la table SCD.

- **End Date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne End Date a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l’option Fixed Year value et saisir une année fictive.

- **Log Active Status** : Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut true et false. Cette colonne permet de repérer facilement le journal actif.

- **Log versions** : Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.

### Advanced settings

#### Debug mode
Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.
**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type *Row Main*. |
| Dynamic settings | Cliquez sur le bouton [*] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table *Dynamic settings* est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue *Basic settings*. Lorsqu’un paramètre dynamique est configuré, la liste *Component List* de la vue *Basic settings* devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez *Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte* à la page 2641 et *Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement* à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*. |
Scénario associé

Pour un scénario associé, consultez tMysqlSCD à la page 2712, ainsi que Scénario : Identifier des modifications de données en utilisant les dimensions à évolution lente (SCD) de type 0 à 3 à la page 2718.
**tMysqlSP**

Ce composant appelle une procédure stockée de base de données MySQL.

Le tMysqlSP permet de centraliser des requêtes multiples ou complexes dans une base de données et de les appeler plus facilement.

### Propriétés du tMysqlSP Standard

Ces propriétés sont utilisées pour configurer le tMysqlSP s’exécutant dans le framework de Jobs Standard.

Le composant tMysqlSP Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

---

#### Basic settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <em>Apply</em>.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <em>Built-in</em> ou <em>Repository</em>.</td>
</tr>
<tr>
<td><em>Built-in</em></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><em>Repository</em></td>
<td>Sélectionnez le fichier des propriétés du composant. Les champs suivants sont alors pré-remplis.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case lorsque vous utilisez le composant <em>tMysqlConnection</em>.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d'authentification sur l'utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <em>Password</em>, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Schema</strong> et <strong>Edit Schema</strong></td>
<td></td>
</tr>
</tbody>
</table>
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

**Repository** : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il est réutilisable. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**SP Name**
Saisissez le nom exact de la Procédure Stockée (SP).

**Is Function / Return result in**
Cochez cette case, si une seule valeur doit être retournée.

Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.

**Parameters**
Cliquez sur le bouton [+] et sélectionnez dans le champ Schema Columns les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure.

Sélectionnez le Type de paramètre :

**IN** : paramètre d'entrée (Input).

**OUT** : paramètre de sortie (Output)/valeur retournée.

**IN OUT** : les paramètres d'entrée doivent être retournés sous forme de valeur, même après modifications via la procédure (fonction).

**RECORDSET** : les paramètres d'entrée doivent être retournés sous forme d'ensemble de valeurs, au lieu d'une valeur unique.

**Remarque :**
Consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620 si vous voulez analyser un ensemble d’enregistrements d’une table de données ou d’une requête SQL.

**Utilisation**

**Règle d’utilisation**

Ce composant est un composant intermédiaire. Il peut également être utilisé comme composant de début. Dans ce cas, seuls les paramètres d’entrée sont autorisés.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Limitation**

La syntaxe de la Procédure Stockée doit correspondre à celle de la base de données.

---

**Scénario : Utiliser le tMysqlSP pour trouver le libellé State à l’aide d’une procédure stockée**

Le Job suivant a pour but de trouver les libellés State en fonction des ID impairs de la colonne State d’une table MySQL à deux colonnes à l’aide d’une procédure stockée.
• Cliquez et déposez les composants suivants dans l’espace de modélisation : tRowGenerator, tMysqlSP et tLogRow.

• Connectez les composants à l’aide d’une connexion de type Row Main.

• Le tRowGenerator est utilisé pour générer des ID impairs. Double-cliquez sur le composant pour lancer l’éditeur.

<table>
<thead>
<tr>
<th>Schema</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column</td>
<td>Key</td>
</tr>
<tr>
<td>ID</td>
<td>☑</td>
</tr>
</tbody>
</table>

• Cliquez sur le bouton [+] pour ajouter une colonne au schéma à générer.

• Cochez la case Key pour définir l’entrée comme étant une clé primaire et définissez le Type en Int (entier).

• Dans le champ Length, la longueur est de 2 chiffres au maximum.

• Utilisez la fonction préétablie appelée sequence mais modifiez les paramètres dans la partie inférieure de la fenêtre.
• Modifiez la valeur (Value) de step : de 1 à 2 pour cet exemple. Mais la valeur de départ (start value) reste 1.

• Dans le champ Number of generated rows, définissez le nombre de lignes à générer à 25, afin que tous les ID impairs de State (il y a 50 Etats) soient générés.

• Cliquez sur OK pour valider la configuration.

• Puis sélectionnez le composant tMysqlSP et paramétrez ses propriétés.

• Dans le champ Property type, sélectionnez l’option Repository puis sélectionnez l’entrée correspondante dans la liste. Les informations de connexion sont remplies automatiquement.

• Sinon, paramétrez les informations de connexion manuellement.

• Cliquez sur Sync Column pour récupérer le schéma généré du composant précédent.
- Puis cliquez sur **Edit Schema** et ajoutez une colonne supplémentaire contenant les libellés des Etats (State) à obtenir en sortie, en plus de l'ID.

- Sélectionnez le type d'encodage dans la liste.

- Dans le champ **SP Name**, saisissez le nom de la procédure comme il a été défini dans la base de données (ici, `getstate`). La procédure à exécuter correspond à :

```
DROP PROCEDURE IF EXISTS `talend`.`getstate` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getstate`(IN pid INT, OUT pstate VARCHAR(50))
BEGIN
    SELECT LabelState INTO pstate FROM us_states WHERE idState = pid;
END $$
```

- Dans la zone **Parameters**, cliquez sur le bouton [+] pour ajouter une ligne à la table.

- Définissez le champ **Column** en **ID**, et le champ **Type** en **IN**, ainsi il sera considéré comme paramètre d'entrée dans la procédure.

- Ajoutez une deuxième ligne et définissez le champ **Column** en **State** et le champ **Type** en **Out**, ainsi il sera considéré comme le paramètre de sortie à retournée.

- Et enfin, paramétrez les propriétés du composant **tLogRow**.

```
<table>
<thead>
<tr>
<th>Basic settings</th>
<th>Schema</th>
<th>Built-In</th>
<th>Edit schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced settings</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic settings</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- Synchronisez le schéma avec celui du composant précédent en cliquant sur le bouton **Sync columns**.

- Et cochez la case **Print values in cells of a table** pour un meilleur confort de lecture.

- Puis enregistrez votre Job et exécutez-le.
La sortie affiche les différents États (State) avec leur ID impair comme défini dans la procédure.

Consultez également Scénario : Insérer des données dans des tables mère/fille à la page 2620 si vous voulez analyser un ensemble d’enregistrements d’une table de données ou d’une requête SQL.

Scénarios associés

Pour des scénarios associés, consultez :

- Scénario : Récupérer des informations personnelles à l’aide d’une procédure stockée à la page 2594.
- Scénario : Vérifier le format de numéros à l’aide d’une procédure stockée à la page 2976.
- Scénario : Exécuter une procédure stockée à l’aide du tMDMSP à la page 2320.
tMysqlTableList

Ce composant liste les noms des tables MySQL d’un jeu de tables donné, à l’aide d’une instruction SELECT se basant sur une clause WHERE.

Le tMysqlTableList effectue une opération d’itération sur toutes les tables d’une base de données, grâce à une connexion MySQL définie.

Propriétés du tMysqlTableList Standard

Ces propriétés sont utilisées pour configurer le tMysqlTableList s’exécutant dans le framework de Jobs Standard.

Le composant tMysqlTableList Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tMysqlConnection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Where clause for table name selection</td>
<td>Saisissez la commande WHERE permettant d’identifier les tables sur lesquelles effectuer l’opération d’itération.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

Global Variables

| CURRENT_TABLE | nom de la table sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères. |
| NB_TABLE | nombre de tables itérées jusqu’à présent. Cette variable est une variable Flow et retourne un entier. |
| ERROR_MESSAGE | message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est |
une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Vous pouvez utiliser ce composant en association avec les autres composants MySQL, notamment avec le tMysqlConnection.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>
**Scénario : Effectuer une opération d’itération sur des tables et en effacer le contenu grâce à un modèle SQL défini par l’utilisateur (SQL Template)**

Le scénario Java suivant décrit un Job à trois composants qui permet d’effectuer une opération d’itération, dans les tables spécifiées d’une base de données MySQL, en utilisant une clause de type WHERE. Le Job permet ensuite d’effacer le contenu de ces tables directement sur le SGBD grâce à un modèle SQL (SQL Template) défini par l’utilisateur.

Pour une utilisation avancée, commencez par créer une connexion à la base de données qui contient les tables que vous voulez vider de leur contenu.

- Dans la vue en arborescence du Repository, développez le nœud Metadata puis cliquez-droit sur DB Connections pour créer une connexion à la base de données de travail et stocker les informations de connexion en local.
  
  Pour plus d’informations sur les métadonnées (metadata), consultez le Guide utilisateur du Studio Talend.

  Sinon, déposez un composant tMysqlConnection dans l’éditeur graphique et renseignez les paramètres de connexion manuellement.

- A partir du Repository, déposez dans l’éditeur graphique la connexion que vous venez de créer. La boîte de dialogue [Components] s’affiche à l’écran.

  Sélectionnez le composant tMysqlConnection puis cliquez sur OK.

  Le composant tMysqlConnection apparaît alors dans l’éditeur graphique et les paramètres de connexion, accessibles depuis la vue Basic settings, sont automatiquement renseignés.

- A partir de la Palette, cliquez-déposez les composants tMysqlTableList et tELT dans l’éditeur graphique.

- Reliez les composants tMysqlConnection et tMysqlTableList à l’aide d’un lien Trigger de type OnSubjobOk.

- Reliez les composants tMysqlTableList et tELT à l’aide d’un lien de type Iterate.

- Double-cliquez sur le tMysqlConnection pour afficher la vue Basic settings du composant si vous voulez en vérifier les paramètres de connexion.
Dans cet exemple, connectez-vous à une base de données MySQL appelée *examples*.

- Dans l'éditeur graphique, double-cliquez sur le composant **tMysqlTableList** pour en afficher les paramètres de base et les définir (vue **Basic settings**).

  ![tMysqlTableList](image)

- Sélectionnez la connexion MySQL appropriée à partir de la liste déroulante **Component list**, dans le cas où vous utilisez plus d'une connexion.

- Renseignez la clause de type *WHERE* dans le champ approprié, en veillant à utiliser la bonne syntaxe, pour effectuer une itération sur la ou les tables dont vous voulez effacer le contenu.

  Dans ce scénario, opérez l’itération sur toutes les tables dont le nom commence par "ex".

- Dans l’éditeur graphique, double-cliquez sur le composant **tELT** pour en afficher les paramètres de base et les définir (vue **Basic settings**).

  ![tELT](image)

- Dans le champ **Database Name**, renseignez le nom de la base de données où se trouvent les tables sur lesquelles vous voulez travailler.

- Sélectionnez la connexion MySQL appropriée à partir de la liste déroulante **Component list**, dans le cas où vous utilisez plus d’une connexion.

- Cliquez dans le champ **Table name** et appuyez sur **Ctrl+Espace** pour atteindre la liste générale des variables.
• A partir de cette liste de variables, sélectionnez `((String)globalMap.get("tMysqlTableList_1_CURRENT_TABLE"))`.

**Comment créer votre propre modèle SQL (SQL Template) :**

• Dans la vue en arborescence du **Repository**, développez successivement les nœuds **SQL Templates** et **MySQL**.

  ![Diagramme d'arborescence du Repository](image)

• Cliquez-droit sur le nœud **UserDefined** et sélectionnez l'option **Create SQLTemplate** à partir de la liste.

  L’assistant **[New SQLTemplate]** s’affiche à l’écran.
• Saisissez un nom pour le nouveau modèle (SQL template) et remplissez les autres champs si nécessaire, puis cliquez sur **Finish** pour fermer l’assistant.

L’éditeur de modèle SQL s’affiche dans l’éditeur graphique.

• Supprimez le code existant et saisissez le code permettant d’exécuter l’action voulue, c’est-à-dire, ici, effacer le contenu de toutes les tables dont les noms commencent par "ex".

```sql
DELETE FROM `<%= "__TABLE-NAME__" %>`;
```

**Remarque :**

Dans un code SQL template, il est impératif d’utiliser le nom exact de la variable, en association avec le paramètre désignant le nom de la table ("__TABLE-NAME__" dans cet exemple). Pour afficher le nom de la variable utilisée, positionnez votre curseur sur le champ **Table Name** de la vue **Basic settings** du composant **tELT**.

• Appuyez sur **Ctrl+S** pour sauvegarder le nouveau modèle que vous venez de définir.

La prochaine étape consiste à ajouter ce nouveau modèle SQL template à la liste de modèles du même type dans le composant **tELT**.

**Comment ajouter votre propre modèle SQL à la liste de modèles du même type :**
• Dans la vue **Component** du composant **tELT**, cliquez sur l'onglet **SQL Templates** pour afficher la liste de modèles **SQLTemplate List**.

![SQL Template List](image)

• Cliquez sur le bouton **Add** et ajoutez deux lignes de modèles SQL template.

• Cliquez sur la première ligne pour afficher un menu déroulante puis cliquez sur la flèche de ce menu pour dérouler la liste des modèles.

![Add button](image)

• Dans la liste, sélectionnez le modèle SQL template que vous venez de créer.

• Assurez-vous que le modèle de la seconde ligne est bien en mode **Commit**.

• Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

Toutes les tables de la base de données MySQL **examples** dont le nom commence par "ex" sont alors vidées de leur contenu.

**Scénario associé**

Pour un scénario associé, consultez **Scénario : Itérer une table de base de données et lister le nom des colonnes de la table** à la page 2611.
tNamedPipeClose

Ce composant ferme un tube nommé à la fin d’un processus.
Le tNamedPipeClose ferme un tube nommé ouvert par le tNamedPipeOpen, à la fin d’un processus.

Propriétés du tNamedPipeClose Standard

Ces propriétés sont utilisées pour configurer le tNamedPipeClose s’exécutant dans le framework de Jobs Standard.
Le composant tNamedPipeClose Standard appartient à la famille File.
Le composant de ce framework est toujours disponible.

Basic settings

| Pipe | Sélectionnez dans la liste un tube nommé existant à fermer. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé pour fermer un tube nommé à la fin d’un Job. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre |
connexion au tube nommé parmi celles prévues dans votre Job.

Lorsqu’un paramètre dynamique est configuré, la case **Pipe** devient inaccessible dans la vue **Basic settings**.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé, consultez **Scénario : Écrire et charger des données via un tube nommé** à la page 2752.
tNamedPipeOpen

Ce composant ouvre un tube nommé pour écrire des données à l'intérieur.

Le tNamedPipeOpen s'utilise dans un processus de communication interne, il ouvre un tube nommé pour écrire des données à l'intérieur.

**Propriétés du tNamedPipeOpen Standard**

Ces propriétés sont utilisées pour configurer le tNamedPipeOpen s'exécutant dans le framework de Jobs Standard.

Le composant tNamedPipeOpen Standard appartient à la famille File.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Name</th>
<th>Saisissez dans le champ le nom du tube nommé.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete if already exist</td>
<td>Cochez cette case si vous ne souhaitez pas dupliquer un tube nommé. Le tube nommé existant sera remplacé.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu'au niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>PIPE_NAME : nom du tube nommé. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PIPE_NATIVE_NAME : nom natif du tube nommé. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>PIPE_OUTPUTSTREAM : flux de sortie du tube nommé. Cette variable est une variable Flow et retourne un object.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début dans un Job effectuant un processus de communication interne.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Limitation</strong></td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez Scénario : Écrire et charger des données via un tube nommé à la page 2752.
tNamedPipeOutput

Ce composant écrit des données dans un tube nommé ouvert.
Le tNamedPipeOutput écrit des données dans un tube nommé ouvert par un tNamedPipeOpen.

Propriétés du tNamedPipeOutput Standard

Ces propriétés sont utilisées pour configurer le tNamedPipeOutput s’exécutant dans le framework de Jobs Standard.
Le composant tNamedPipeOutput Standard appartient à la famille File.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use existing pipe connection</th>
<th>Cochez cette case afin d’utiliser un tube nommé existant dans la liste Pipe component, ou décochez cette case afin de spécifier un tube nommé dans le champ Pipe name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipe component</td>
<td>Sélectionnez dans la liste un composant NamedPipe existant.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Cette case est disponible uniquement lorsque vous cochez la case Use existing pipe connection.</td>
</tr>
<tr>
<td>Pipe name</td>
<td>Saisissez dans le champ le nom d’un tube nommé existant.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Cette case s’affiche uniquement lorsque vous décochez la case Use existing pipe connection.</td>
</tr>
<tr>
<td>Row separator</td>
<td>Chaîne (ex : &quot;\n&quot; sous Unix) séparant les lignes dans le fichier de sortie.</td>
</tr>
<tr>
<td>Field separator</td>
<td>Caractère, chaîne ou expression régulière pour séparer les champs du fichier de sortie.</td>
</tr>
<tr>
<td>CSV options</td>
<td>Cochez cette case afin de prendre en compte tous les paramètres spécifiques aux fichiers CSV, en particulier les paramètres Escape char et Text enclosure.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
</tbody>
</table>
- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

|---|---|

**Delete pipe if it exists**

Cochez cette case si vous ne souhaitez pas dupliquer un tube nommé. Le tube nommé existant sera remplacé.

**Advanced settings**

<table>
<thead>
<tr>
<th>Boolean type</th>
<th>Sélectionnez un type de booléen dans la liste.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>NB_LINE</strong> : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <strong>Flow</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>PIPE_NAME</strong> : nom du tube nommé. Cette variable est une variable <strong>Flow</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>PIPE_NATIVE_NAME</strong> : nom natif du tube nommé. Cette variable est une variable <strong>Flow</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>PIPE_OUTPUTSTREAM</strong> : flux de sortie du tube nommé. Cette variable est une variable <strong>Flow</strong> et retourne un object.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé connecté à un autre composant dans un sous-job lisant des données d’une source.</th>
</tr>
</thead>
</table>
| Dynamic settings   | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion au tube nommé parmi celles prévues dans votre Job.  

La table **Dynamic settings** n’est disponible que si la case **Use existing pipe connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Pipe component** devient inaccessible dans la vue **Basic settings**.  

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**. |
| Limitation          | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com). |

### Scénario : Écrire et charger des données via un tube nommé

Le scénario suivant permet de créer un Job écrivant des données dans un tube nommé ouvert et affichant ces données dans la console.
Déposer et relier les composants

Procédure
1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : tNamedPipeOpen, tParallelize, tNamedPipeClose, tFileInputDelimited, tSleep, tLogRow, tRowGenerator et tNamedPipeOutput.
2. Connectez le tNamedPipeOpen au tParallelize à l’aide d’un lien Trigger > OnSubjobOk.
3. Reliez le composant tParallelize au tFileInputDelimited à l’aide d’un lien Trigger > Parallelize.
4. Connectez le tParallelize au composant tSleep à l’aide d’un lien Trigger > Parallelize.
5. Reliez le tFileInputDelimited au tLogRow à l’aide d’un lien Row > Main.
6. Connectez le composant tParallelize au tNamedPipeClose à l’aide d’un lien Trigger > Synchronize (Wait for all).
7. Reliez le tSleep au composant tRowGenerator à l’aide d’un lien Trigger > OnComponentOk.
8. Connectez le tRowGenerator au tNamedPipeOutput à l’aide d’un lien Row > Main.

Configurer les composants

Configurer le composant d’entrée

Procédure
1. Double-cliquez sur le tNamedPipeOpen afin de définir ses propriétés dans la vue Basic settings.
   Dans le champ Name, saisissez le nom d’un tube nommé et sélectionnez Delete if already exist afin de ne pas dupliquer ce tube nommé.
2. Double-cliquez sur le **tParallelize** afin d’afficher sa vue **Basic settings**.

   Sélectionnez **end of all subjobs** dans la liste **Wait for**.

   Dans le champ **Sleep Duration**, saisissez 100 pour configurer la durée de pause.

3. Double-cliquez sur le **tFileInputDelimited** afin d’afficher sa vue **Basic settings**.

   Dans le champ **File name/Stream**, saisissez l’expression suivante pour utiliser le nom du tube nommé existant défini dans la vue **Basic settings** du **tNamedPipeOpen** :

   ```java
   ((String)globalMap.get("tNamedPipeOpen_1_PIPE_NATIVE_NAME"))
   ```

4. Cliquez sur le bouton [...] à côté du champ **Edit schema**.
5. Cliquez sur le bouton [+] pour ajouter trois colonnes au tFileInputDelimited. Renommez-les id, first_name et last_name puis configurez le Type de la colonne id à Integer. Laissez les autres paramètres tels qu’ils sont.

6. Cliquez sur OK afin de sauvegarder le schéma.

7. Laissez les autres paramètres de la vue Basic settings tels qu’ils sont.

8. Double-cliquez sur le tSleep et saisissez 1 dans le champ Pause (in seconds).

9. Double-cliquez sur le tRowGenerator pour afficher sa vue Basic settings.

10. Cliquez sur RowGenerator Editor pour configurer le schéma.

11. Cliquez sur le bouton [+] pour ajouter trois colonnes au tRowGenerator. Renommez ces trois colonnes respectivement id, first_name et last_name puis configurez le Type de la colonne id à Integer. Laissez les autres types tels qu’ils sont par défaut.

12. Sélectionnez sequence dans la liste Functions pour la colonne id.

13. Sélectionnez getFirstName dans la liste Functions pour la colonne first_name.
14. Sélectionnez `TalendDataGenerator.getLastName` dans la liste **Functions** pour la colonne **Column last_name**.

15. Sélectionnez `id` dans le champ **Value** sous l'onglet **Function parameters** pour **sequence identifier**, 1001 pour **start value** et 1 pour **step**.

16. Cliquez sur **OK** pour sauvegarder les modifications.

**Configurer le composant de sortie**

**Procédure**

1. Double-cliquez sur le composant **tNamedPipeOutput** pour afficher sa vue **Basic settings**.

2. Cochez la case **Use existing pipe connection** et sélectionnez **tNamedPipeOpen_1** dans la liste **Pipe component**.

3. Cochez la case **Delete pipe if it exists** pour ne pas dupliquer le tube nommé.

4. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.

5. Laissez les autres paramètres tels qu’ils sont.

6. Double-cliquez sur le **tLogRow** pour afficher sa vue **Basic settings**.

7. Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.

8. Sélectionnez **Table** dans la zone **Mode**.
9. Double-cliquez sur le `tNamedPipeClose` afin d' afficher sa vue `Basic settings`.
10. Sélectionnez le composant `tNamedPipeOpen_1` dans la liste `Pipe`.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

```
Starting job namedPipe_Sample_01 at 18:13 19/05/2011.
[statistics] connecting to socket on port 3842
[statistics] connected
   tLogRov_1
   [id | first_name | last_name]
   [----------|----------|----------]
   1001 | Bill | Cleveland
[statistics] disconnected
Job namedPipe_Sample_01 ended at 18:13 19/05/2011. [exit code=0]
```

**Résultats**

Les données écrites dans le tube nommé sont affichées dans la console.
tNeo4jBatchOutput

Ce composant reçoit des données du composant précédent et écrit ces données dans une base de données Neo4j locale.
Le tNeo4jBatchOutput est utilisé pour écrire des nœuds dans une base de données Neo4j en se basant sur l’index défini.

Propriétés du tNeo4jBatchOutput Standard

Ces propriétés sont utilisée pour configurer le composant tNeo4jBatchOutput s’exécutant dans le framework de Jobs Standard.
Le composant tNeo4jBatchOutput Standard appartient aux familles Big Data et Databases.
Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.
Ce composant supporte uniquement les versions 3.2.X de Neo4j et ne supporte pas le mode distant. Ne réutilisez pas de connexion à des versions autres que V3.2.X configurée dans un composant tNeo4jConnection et ne cochez pas la case Remote server dans les options du tNeo4jConnection.
N’utilisez pas de version 2.X.X et 3.X.X dans un même Job. Sinon, des conflits de classes peuvent survenir. |
| Database path | Spécifiez le chemin vers le répertoire qui contiendra vos fichiers de données. Si le répertoire spécifié n’existe pas, il sera créé.
Ce champ s’affiche uniquement lorsque vous ne cochez pas la case Use an existing connection. |
| Shutdown after job | Cochez cette case pour arrêter la connexion à la base de données Neo4j lorsqu’aucune opération sur Neo4j n’est effectuée après le composant courant.
D’une manière alternative, vous pouvez utiliser le tNeo4jClose pour arrêter la base de données.
Cela permet d’éviter des erreurs comme “Id file not properly shutdown” lors de la prochaine exécution de Jobs impliquant Neo4j.
Cette case est disponible uniquement lorsque la case Use an existing connection est cochée. |
| Field that contains the label list | Sélectionnez la colonne du schéma d’entrée défini dans les composants précédents afin de fournir des libellés pour les nœuds à créer. |
Lorsque les données d’entrée fournissent plusieurs libellés pour un nœud, séparez ces libellés par un point-virgule (;) dans les composants précédents.

### Index name
Saisissez, entre guillemets doubles, le nom de l’index à créer pour les nœuds. Cet index est utilisé pour stocker les identifiants des nœuds et permettre de créer les relations.

### Import identifier
 Sélectionnez la colonne du schéma d’entrée défini dans les composants précédents afin de fournir les identifiants des nœuds à créer. Ces identifiants sont stockés dans l’index à créer et doivent être uniques à travers l’index.
Assurez-vous que la colonne à utiliser contient des valeurs uniques.

### Save the identifier
Cochez cette case afin de sauvegarder les identifiants techniques (ID des graphes natifs).
Pour plus d’informations concernant les différents types d’identifiants d’entité Neo4j, consultez la documentation Neo4j : Entity identifier (en anglais).

### Number of index elements in memory
Saisissez, sans guillemet double, la taille du cache utilisé pour conserver les éléments de l’index en mémoire. Cette valeur doit être suffisante pour conserver en mémoire tous les éléments de l’index.
Pour plus d’informations, consultez la documentation Neo4j : Memory tuning (en anglais).

### Advanced settings

#### Neo4j configuration
Ajoutez des paramètres à la table pour configurer la base de données à créer.
Pour plus d’informations, consultez la documentation Neo4j : Configuration settings (en anglais).
Lorsque vous saisissez des valeurs, utilisez la syntaxe fournie par les exemples avec les noms des colonnes de la table Nodes files.

#### tStatCatcher Statistics
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

#### Variables globales
**NB_LINE**: nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

**ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est utilisé en tant que composant de sortie et nécessite un lien d’entrée.</th>
</tr>
</thead>
</table>
tNeo4jBatchOutputRelationship

Ce composant reçoit des données du composant précédent et écrit en masse des relations dans une base de données Neo4j locale.

**Propriétés du tNeo4jBatchOutputRelationship Standard**

Ces propriétés sont utilisées pour configurer le tNeo4jBatchOutputRelationship s’exécutant dans le framework de Jobs Standard.

Le composant tNeo4jBatchOutputRelationship Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Use existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Ce composant supporte uniquement les versions 3.2.X de Neo4j et ne supporte pas le mode distant. Ne réutilisez pas de connexion à des versions autres que V3.2.X configurée dans un composant tNeo4jConnection et ne cochez pas la case Remote server dans les options du tNeo4jConnection.

N’utilisez pas de version 2.X.X et 3.X.X dans un même Job. Sinon, des conflits de classes peuvent survenir. |
|---|---|
| Database path | Spécifiez le répertoire qui contiendra vos fichiers de données.

Ce champ s’affiche uniquement lorsque vous ne cochez pas la case Use an existing connection. |
| Shutdown after job | Cochez cette case pour arrêter la connexion à la base de données Neo4j lorsqu’aucune opération sur Neo4j n’est effectuée après le composant courant.

D’une manière alternative, vous pouvez utiliser le tNeo4jClose pour arrêter la base de données.

Cela permet d’éviter des erreurs comme “Id file not properly shutdown” lors de la prochaine exécution de Jobs impliquant Neo4j.

Cette case est disponible uniquement lorsque la case Use an existing connection est cochée. |
| Field for relationship types | Sélectionnez la colonne du schéma d’entrée défini dans les composants précédents afin de fournir des types pour les relations à créer. |
| Direction of the relationship | Sélectionnez la direction de la relation à créer : |
- **Outgoing** : la relation démarre du nœud de début et termine au nœud de fin.
- **Incoming** : la relation démarre du nœud de fin et termine au nœud de début.

### Start node of the relationship
Définissez le nœud de début de chaque relation, à l'aide de l'identifiant de nœud :
- **Name of the batch index** : sélectionnez le composant tNeo4jBatchOutput utilisé pour créer les nœuds de début. Le nom de l'index de ces nœuds est récupéré de ce composant.
- **Field name for the batch index** : sélectionnez la colonne du schéma d'entrée défini dans les composants précédents afin de fournir le nom du nœud de début de chaque relation à créer.

### End node of the relationship
Définissez le nœud de fin de chaque relation, à l'aide de l'identifiant de nœud :
- **Name of the batch index** : sélectionnez le composant tNeo4jBatchOutput utilisé pour créer les nœuds de fin. Le nom de l'index de ces nœuds est récupéré de ce composant.
- **Field name for the batch index** : sélectionnez la colonne du schéma d'entrée défini dans les composants précédents afin de fournir le nom du nœud de fin de chaque relation à créer.

### Die on error
Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient.
Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.

### Advanced settings

#### Neo4j configuration
Ajoutez des paramètres à la table pour configurer la base de données à créer.
Pour plus d'informations, consultez la documentation Neo4j : Configuration settings (en anglais).
Lorsque vous saisissez des valeurs, utilisez la syntaxe fournie par les exemples avec les noms des colonnes de la table Nodes files.

#### tStatCatcher Statistics
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.

### Variables globales

#### Variables globales
**NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | Ce composant est utilisé en tant que composant de sortie et nécessite un lien d’entrée. |

**Écrire dans Neo4j des informations relatives à des acteurs et films avec une relation hiérarchique**

Dans ce scénario, les composants Batch Neo4j sont utilisés pour importer des données relatives à des acteurs et films à partir de deux fichiers CSV dans une base de données Neo4j locale et créer une relation pour les données, en se basant sur un autre fichier CSV décrivant les rôles des acteurs au sein des films.

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Les composants Batch Neo4j fournis par Talend supportent l’écriture en masse uniquement dans une base de données Neo4j locale. Ils ne peuvent être utilisés ni avec les versions de Neo4j antérieures à la V3.2.X, ni avec des composants Neo4j utilisant l’une de ces versions de Neo4j.

Les composants à utiliser sont :

- un tNeo4jConnection : pour ouvrir la connexion Neo4j à réutiliser,
- trois tFileInputDelimited : pour lire les informations d’entrée relatives aux acteurs et films,
- deux tNeo4jBatchOutput : pour écrire les informations relatives aux films et acteurs dans la base de données Neo4j connectée,
- un tNeo4jBatchOutputRelationship : pour créer la relation entre les acteurs et les films,
- un tNeo4jBatchSchema : pour créer une contrainte d’unicité sur les nœuds dans le base de données.
Créer le Job Batch Neo4j

Procédure

1. Assurez-vous que le statut de votre service et de votre console Neo4j est bien stop.
   Si vous utilisez une invite de commande pour gérer Neo4j, vous pouvez utiliser neo4j status pour vérifier le statut ; si vous avez installé l’application Neo4j, vous pouvez vérifier directement dans cette application.

2. Dans le Repository de la perspective Integration, créez un Job et ajoutez les composants à utiliser en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.

3. Reliez le premier composant tFileInputDelimited au premier tNeo4jBatchOutput à l’aide d’un lien Row > Main. Ce sous-job importe les données des acteurs dans la base de données Neo4j.

4. Reliez le tFileInputDelimited au second tNeo4jBatchOutput à l’aide d’un lien Row > Main. Ce sous-job importe les données des films dans la base de données Neo4j.

5. Reliez le troisième tFileInputDelimited au tNeo4jBatchOutputRelationship à l’aide d’un lien Row > Main. Ce sous-job crée une relation entre les acteurs et les films.


Configurer la connexion Neo4j à réutiliser

Procédure

1. Double-cliquez sur le composant tNeo4jConnection pour ouvrir sa vue Basic settings.
2. Dans la liste **DB Version**, sélectionnez **Neo4J 3.2.X**.

3. Assurez-vous que la case **Use a remote server** est décochée car les composants Batch fonctionnent uniquement en mode local.

4. Dans le champ **Database path**, saisissez le chemin d'accès au fichier ou parcourez votre système jusqu’au fichier de la base de données.

### Écrire en masse les données des acteurs dans Neo4j

#### Procédure

1. Double-cliquez sur le premier **tFileInputDelimited** pour ouvrir sa vue **Component**.

2. Dans le champ **File name/Stream**, saisissez le chemin d'accès ou parcourez votre système jusqu’au fichier CSV décrivant les ID, les noms et les libellés des acteurs, à utiliser dans Neo4j.

Le fichier CSV d’entrée utilisé dans cet exemple se présente comme suit :

```plaintext
keanu,"Keanu Reeves",Actor  
laurence,"Laurence Fishburne",Actor  
carrieanne,"Carrie-Anne Moss",Actor  
```

Les guillemets doubles autour du nom des acteurs ne sont pas obligatoires.

3. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur de schéma et définissez le schéma en vous basant sur la structure du fichier d’entrée.

Dans cet exemple, les colonnes sont **id, name et label**, toutes de type **String**.

5. Dans le champ Field separator, saisissez une virgule (,) pour remplacer le point-virgule par le point-virgule (;) par défaut.

6. Double-cliquez sur le premier tNeo4jBatchOutput pour ouvrir sa vue Component.

7. Cochez la case Use an existing connection pour réutiliser la connexion à la base de données Neo4j ouverte par le composant tNeo4jConnection.

8. Vérifiez que la case Shutdown after Job est décochée.

9. Dans la liste déroulante Field that contains the label list, sélectionnez la colonne fournissant les libellés.

10. Dans le champ Index name, saisissez le nom de l’index à créer pour les nœuds.

11. Dans la liste déroulante Import identifier, sélectionnez la colonne fournissant les ID.

Écrire en masse les données des films dans Neo4j

Procédure

1. Double-cliquez sur le second tFileInputDelimited pour ouvrir sa vue Component.
2. Dans le champ **File name/Stream**, saisissez le chemin d'accès ou parcourez votre système jusqu'au fichier CSV décrivant les ID, noms, années de sortie et libellés à utiliser dans Neo4j.

Le fichier CSV d’entrée se présente comme suit, dans cet exemple :

```
tt0133093, "The Matrix", 1999, Movie
tt0234215, "The Matrix Reloaded", 2003, Movie; Sequel
tt0242653, "The Matrix Revolutions", 2003, Movie; Sequel
```

Les guillemets doubles autour des noms de films ne sont pas obligatoires.

3. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur et définissez le schéma d’entrée en vous basant sur la structure du fichier d’entrée.

Dans cet exemple, les colonnes sont **id**, **title**, **released** et **label**.


5. Dans le champ **Field separator**, saisissez une virgule (,) pour remplacer le point-virgule par défaut (;).

6. Double-cliquez sur le second **tNeo4jBatchOutput** pour ouvrir sa vue **Component**.
7. Cochez la case **Use an existing connection** pour réutiliser la connexion à la base de donnée Neo4j ouverte par le composant tNeo4jConnection.

8. Vérifiez que la case **Shutdown after Job** est décochée.

9. Dans la liste déroulante **Field that contains the label list**, sélectionnez la colonne qui fournit le libellé.

10. Dans le champ **Index name**, saisissez le nom de l’index à créer pour les nœuds.

11. Dans la liste déroulante **Import identifier**, sélectionnez la colonne fournissant les ID.

**Créer des relations en masse**

**Procédure**

1. Double-cliquez sur le troisième tFileInputDelimited pour ouvrir sa vue Component.
2. Dans le champ **File name/Stream**, saisissez le chemin d'accès au fichier ou parcourez votre système jusqu'au fichier CSV décrivant les relations acteur-film.

Le fichier d'entrée CSV utilisé dans cet exemple se présente comme suit :

<table>
<thead>
<tr>
<th>Acteur</th>
<th>Rôle</th>
<th>ID Film</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keanu</td>
<td>Neo</td>
<td>tt0133093</td>
<td>ACTED_IN</td>
</tr>
<tr>
<td>Keanu</td>
<td>Neo</td>
<td>tt0234215</td>
<td>ACTED_IN</td>
</tr>
<tr>
<td>Keanu</td>
<td>Neo</td>
<td>tt0242653</td>
<td>ACTED_IN</td>
</tr>
<tr>
<td>Laurence</td>
<td>Morpheus</td>
<td>tt0133093</td>
<td>ACTED_IN</td>
</tr>
<tr>
<td>Laurence</td>
<td>Morpheus</td>
<td>tt0234215</td>
<td>ACTED_IN</td>
</tr>
<tr>
<td>Laurence</td>
<td>Morpheus</td>
<td>tt0242653</td>
<td>ACTED_IN</td>
</tr>
<tr>
<td>Carrieanne</td>
<td>Trinity</td>
<td>tt0133093</td>
<td>ACTED_IN</td>
</tr>
<tr>
<td>Carrieanne</td>
<td>Trinity</td>
<td>tt0234215</td>
<td>ACTED_IN</td>
</tr>
<tr>
<td>Carrieanne</td>
<td>Trinity</td>
<td>tt0242653</td>
<td>ACTED_IN</td>
</tr>
</tbody>
</table>

Les guillemets doubles entourant les noms de rôles ne sont pas obligatoires. La valeur **ACTED_IN** est un type de relation personnalisé expliquant la relation entre les acteurs et les films.

3. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur de schéma et définir le schéma d’entrée en vous basant sur la structure du fichier d’entrée.

Dans cet exemple, les colonnes sont from, role, to et type.


5. Dans le champ **Field separator**, saisissez une virgule (,) pour remplacer le point-virgule par défaut (;).

6. Double-cliquez sur le composant **tNeo4jBatchOutputRelationship** pour ouvrir sa vue **Component**.
7. Cochez la case **Use an existing connection** pour réutiliser la connexion à la base de données Neo4j ouverte par le composant tNeo4jConnection.

8. Vérifiez que la case **Shutdown after Job** est décochée.

9. Dans la liste déroulante **Field for relationship type**, sélectionnez la colonne fournissant les types de relations.

10. Dans la liste déroulante **Direction of the relationship**, sélectionnez **Outgoing**.

11. Dans la zone **Start node of the relationship**, sélectionnez le composant **tNeo4jBatchOutput** fournissant l’index des nœuds de début, l’index `asActors` dans cet exemple, provenant du premier tNeo4jBatchOutput. Dans la liste déroulante **Field name for the batch index**, sélectionnez la colonne fournissant les noms d’acteurs comme nœuds de début.

12. Répétez cette action dans la zone **End node of the relationship** pour sélectionner l’index `asMovie` provenant du second tNeo4jBatchOutput et sélectionnez la colonne fournissant les noms de films comme nœuds de fin.

**Ajouter des contraintes d’unicité sur les nœuds**

**Procédure**

1. Double-cliquez sur le composant **tNeo4jBatchSchema** pour ouvrir sa vue **Component**.
2. Cochez la case **Use an existing connection** pour réutiliser la connexion à la base de données Neo4j ouverte par le tNeo4jConnection.

3. Cochez la case **Shutdown after Job** pour fermer correctement la connexion après l’exécution.

4. Dans la table **Schema definition**, ajoutez deux lignes en cliquant deux fois sur le bouton **[+]** :
   a) Dans la colonne **Schema type**, sélectionnez **Node property is unique** pour les deux lignes, pour ajouter des contraintes d’unicité aux nœuds dans Neo4j.
   b) Dans la colonne **For node with Label**, saisissez, entre guillemets doubles, **Actor** et **Movie**, respectivement, qui sont les libellés utilisés par les nœuds des acteurs et les nœuds des films. Ce que vous saisissez ici doit être identique aux libellés précédemment utilisés lors de la création de ces nœuds.
   c) Dans la colonne **On property**, saisissez, entre guillemets doubles, les propriétés du nœud sur lequel ajouter des contraintes d’unicité. Pour les nœuds des acteurs, saisissez **name** et, pour les nœuds des films, saisissez **title**. Les valeurs que vous saisissez ici doivent être identiques aux noms des colonnes précédemment définies pour fournir les noms des films et des acteurs pour les nœuds à créer par les tNeo4jBatchOutput.

5. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job et appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour exécuter le Job.

**Résultats**

Une fois le Job exécuté correctement, vérifiez les résultats dans votre navigateur Neo4j :
"The Matrix" "The Matrix"
"Carrie"
"Lauren"
"Keanu Reeves"

ACTED_IN <id>: 2 role: "Neo"
tNeo4jBatchSchema

Defines the schema of a local Neo4j database.

Propriétés du tNeo4jBatchSchema Standard

Ces propriétés sont utilisées pour configurer le tNeo4jBatchSchema s'exécutant dans le framework de Jobs Standard.

Le composant tNeo4jBatchSchema Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.

Ce composant supporte uniquement les versions 3.2.X de Neo4j et ne supporte pas le mode distant. Ne réutilisez pas de connexion à des versions autres que V3.2.X configurée dans un composant tNeo4jConnection et ne cochez pas la case Remote server dans les options du tNeo4jConnection.

N'utilisez pas de version 2.X.X et 3.X.X dans un même Job. Sinon, des conflits de classes peuvent survenir. |
|---|---|

| Database path | Spécifiez le répertoire qui contiendra vos fichiers de données. Si le répertoire spécifié n'existe pas, il sera créé.

Ce champ s'affiche uniquement lorsque vous ne cochez pas la case Use an existing connection. |
|---|---|

| Shutdown after job | Cochez cette case pour arrêter la connexion à la base de données Neo4j lorsqu'aucune opération sur Neo4j n'est effectuée après le composant courant.

D'une manière alternative, vous pouvez utiliser le tNeo4jClose pour arrêter la base de données.

Cela permet d'éviter des erreurs comme "ld file not properly shutdown" lors de la prochaine exécution de Jobs impliquant Neo4j.

Cette case est disponible uniquement lorsque la case Use an existing connection est cochée. |
|---|---|

| Schema definition | Ajoutez des index et des contraintes de propriétés uniques pour définir le schéma :

- **Schema type** : choisissez si le type de schéma est un index (Index on node property) ou une contrainte (Node property is unique) à définir. |
|---|---|
For node with label : saisissez, entre guillemets doubles, le libellé d'un nœud.

On property : saisissez, entre guillemets doubles, la propriété d'un nœud.

Advanced settings

Neo4j configuration
Ajoutez des paramètres à la table pour configurer la base de données à créer.
Pour plus d’informations, consultez la documentation Neo4j : Configuration settings (en anglais).
Lorsque vous saisissez des valeurs, utilisez la syntaxe fournie par les exemples avec les noms des colonnes de la table Nodes files.

tStatCatcher Statistics
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.

Global Variables

Variables globales

| NB_LINE | nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.
| ERROR_MESSAGE | message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation
Ce composant est utilisé en standalone dans un sous-job pour écrire un schéma de graphe Neo4j.
tNeo4jClose

Ce composant ferme une connexion active à une base de données Neo4j en mode embarqué.

Propriétés du tNeo4jClose Standard

Ces propriétés sont utilisées pour configurer le tNeo4jClose s’exécutant dans le framework de Jobs Standard.

Le composant tNeo4jClose Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Connection | Sélectionnez la connexion active à la base de données Neo4j que vous souhaitez fermer. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau des composants. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Neo4j, notamment le tNeo4jConnection. |
Scénario associé

Pour des scénarios utilisant un **tNeo4jClose**, consultez :

- Scénario : Ecrire des informations concernant des acteurs et des films dans Neo4j avec des relations hiérarchiques à la page 2802,
- Scénario 1 : Créer des nœuds avec un libellé à l’aide d’une requête Cypher à la page 2815, et
- Scénario 2 : Importer des données dans une base de données Neo4j à partir d’un fichier CSV à l’aide d’une requête Cypher à la page 2819.
tNeo4jConnection

Ce composant ouvre une connexion à une base de données Neo4j à réutiliser par d’autres composants Neo4j.
Le tNeo4jConnection démarre la base de données en mode embarqué et vérifie la disponibilité du serveur, en mode REST.

Propriétés du tNeo4jConnection Standard

Ces propriétés sont utilisées pour configurer le tNeo4jConnection s’exécutant dans le framework de Jobs Standard.
Le composant tNeo4jConnection Standard appartient aux familles Big Data et Databases.
Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB version</td>
<td>Sélectionnez la version de Neo4j que vous utilisez. Après avoir sélectionné la version de la base de données, il vous est demandé d’installer les fichiers Jar contenant les pilotes correspondant à la base de données. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
<tr>
<td>Remote server</td>
<td>Cochez cette case si vous utilisez un serveur distant Neo4j et spécifiez l’URL racine dans le champ Server URL.</td>
</tr>
<tr>
<td>Server URL</td>
<td>Spécifiez l’URL racine. Ce champ est disponible uniquement si la case Use a remote server est cochée.</td>
</tr>
<tr>
<td>Read only</td>
<td>Cochez cette case si vous souhaitez utiliser la base de données embarquée en mode lecture seule. Cette option est utile si une application est déjà dédiée à la base de données. Cette case est disponible uniquement si vous utilisez une version 2.X.X. Avertissement : Il est recommandé de ne pas utiliser ce mode lorsque votre Job comprend au moins un composant de sortie Neo4j, comme le tNeo4jOutput, le tNeo4jOutputRelationship ou le tNeo4jRow.</td>
</tr>
<tr>
<td>Database path</td>
<td>Si vous utilisez Neo4j en mode embarqué, spécifiez le chemin d’accès au fichier de données.</td>
</tr>
</tbody>
</table>
Ce champ est disponible uniquement si la case **Use a remote server** n’est pas cochée.

### Advanced settings

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau des composants. |

### Variables globales

| **Variables globales** | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

### Utilisation

| **Règle d’utilisation** | Ce composant est généralement utilisé avec d’autres composants Neo4j, notamment le **tNeo4jClose**. |

### Scénario associé

Pour des scénarios utilisant un **tNeo4jConnection**, consultez :

- **Scénario** : Ecrire des informations concernant des acteurs et des films dans Neo4j avec des relations hiérarchiques à la page 2802,

- **Scénario 1** : Créer des nœuds avec un libellé à l’aide d’une requête Cypher à la page 2815, et

- **Scénario 2** : Importer des données dans une base de données Neo4j à partir d’un fichier CSV à l’aide d’une requête Cypher à la page 2819.
tNeo4jImportTool

Ce composant utilise l’outil Neo4j Import Tool pour créer une base de données Neo4j et importer de grands volumes de données en masse depuis des fichiers CSV vers cette base de données.

Propriétés du tNeo4jImportTool Standard

Ces propriétés sont utilisées pour configurer le composant tNeo4jImportTool s’exécutant dans le framework de Jobs Standard.

Le composant tNeo4jImportTool Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Neo4j database folder</th>
<th>Saisissez le chemin du, ou parcourez votre système jusqu’au répertoire dans lequel vous souhaitez que le tNeo4jImportTool crée une base de données Neo4j. Le nom que vous saisissez dans le chemin est utilisé pour nommer cette base de données.</th>
</tr>
</thead>
</table>

Nodes files

Renseignez cette table pour fournir les fichiers CSV contenant les informations relatives aux nœuds à importer.

- Path to CSV file : saisissez, entre guillemets doubles, le chemin vers le fichier CSV à lire.
- CSV headers : saisissez, entre guillemets doubles, les en-têtes utilisés dans ce fichier CSV.

Lorsque vous saisissez des valeurs, utilisez la syntaxe fournie par les exemples données avec les noms des colonnes dans la table Nodes files.

Relationship files

Renseignez cette table afin de fournir les fichiers CSV contenant les informations relatives aux relations à importer.

- Path to CSV file : saisissez, entre guillemets doubles, le chemin vers le fichier CSV à lire.
- CSV headers : saisissez, entre guillemets doubles, les en-têtes utilisés dans ce fichier CSV.

Lorsque vous saisissez des valeurs, utilisez la syntaxe fournie par les exemples données avec les noms des colonnes dans la table Nodes files.

Advanced settings

| Neo4j configuration | Ajoutez des paramètres à la table pour configurer la base de données à créer. Pour plus d’informations, consultez la documentation Neo4j : [Configuration settings](#) (en anglais). |
Lorsque vous saisissez des valeurs, utilisez la syntaxe fournie par les exemples avec les noms des colonnes de la table Nodes files.

Import configuration

Ajoutez des paramètres à la table afin de configurer Neo4j Import Tool.
Pour plus d'informations, consultez la documentation Neo4j : Import Tool usage (en anglais).
Lorsque vous saisissez des valeurs, utilisez la syntaxe fournie par les exemples données avec les noms des colonnes dans la table Nodes files.

tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

Variables globales

Variables globales

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant est utilisé en standalone dans un sous-job pour charger des données en masse de fichiers CSV ayant été créés avec une structure spécifique.
Pour plus d’informations concernant la structure CSV requise par Neo4j Import Tool, consultez Import (en anglais).
Vous pouvez utiliser un tFileOutputDelimited pour créer les fichiers CSV à utiliser. Si vous faites ceci, gardez à l’esprit la configuration suivante :
• Line separator : "\n"
• Field separator : ", ,"
• N’incluez pas d’en-tête.
• Encodage du fichier : UTF-8.
tNeo4jInput

Ce composant lit des données de Neo4j et les envoie dans un flux de sortie Talend.

Le tNeo4jInput lit des données Neo4j selon une requête Cypher vous permettant d'effectuer des transformations ou des traitements dans le reste du Job.

Propriétés du tNeo4jInput Standard

Ces propriétés sont utilisées pour configurer le tNeo4jInput s'exécutant dans le framework de Jobs Standard.

Le composant tNeo4jInput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>DB version</td>
<td>Sélectionnez la version de Neo4j que vous utilisez.</td>
</tr>
<tr>
<td></td>
<td>Cette liste n’est pas affichée lorsque la case Use an existing connection est cochée.</td>
</tr>
<tr>
<td></td>
<td>Après avoir sélectionné la version de la base de données, il vous est demandé d’installer les fichiers Jar contenant les pilotes correspondant à la base de données.</td>
</tr>
<tr>
<td>Shutdown after job</td>
<td>Cochez cette case pour arrêter la connexion à la base de données Neo4j lorsqu’aucune opération sur Neo4j n’est effectuée après le composant courant.</td>
</tr>
<tr>
<td></td>
<td>D’une manière alternative, vous pouvez utiliser le tNeo4jClose pour arrêter la base de données.</td>
</tr>
<tr>
<td></td>
<td>Cela permet d’éviter des erreurs comme “Id file not properly shutdown” lors de la prochaine exécution de Jobs impliquant Neo4j.</td>
</tr>
<tr>
<td></td>
<td>Cette case est disponible uniquement lorsque la case Use an existing connection est cochée.</td>
</tr>
<tr>
<td>Remote server</td>
<td>Cochez cette case si vous utilisez un serveur distant Neo4j et spécifiez l’URL racine dans le champ Server URL.</td>
</tr>
<tr>
<td>Database path</td>
<td>Si vous utilisez Neo4j en mode embarqué, spécifiez le chemin d'accès au fichier de données.</td>
</tr>
</tbody>
</table>
Ce champ est disponible si vous ne cochez pas la case Use an existing connection ou Remote server.

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
</tr>
</thead>
</table>
| Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.  

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

<table>
<thead>
<tr>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisissez votre requête Cypher entre guillemets doubles, avec des paramètres de retour correspondant à la table de mapping.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Renseignez cette table afin de spécifier la ou les colonne(s) à extraire, ainsi que la ou les famille(s) de colonnes correspondantes. La colonne Column de cette table est automatiquement renseignée une fois le schéma défini. Renseignez les champs Return parameter avec les paramètres de retour permettant de mapper les propriétés du nœud dans la requête Cypher et les colonnes du schéma.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Advanced settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
</tr>
<tr>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables globales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables globales</td>
</tr>
</tbody>
</table>
| **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.  

**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

| Règle d’utilisation | Ce composant nécessite un lien de sortie. |

### Scénario associé

Pour un scénario associé, consultez **Scénario 1 : Écrire des données dans une base de données Neo4j et lire des données spécifiques de cette base de données** à la page 2788.
tNeo4jOutput

Ce composant reçoit des données du composant précédent et les écrit dans Neo4j.

Le tNeo4jOutput est utilisé pour écrire des données dans une base de données Neo4j et/ou mettre à jour ou supprimer des entrées dans la base de données, selon l’index défini.

Propriétés du tNeo4jOutput Standard

Ces propriétés sont utilisées pour configurer le tNeo4jOutput s’exécutant dans le framework de Jobs Standard.

Le composant tNeo4jOutput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>DB version</td>
<td>Sélectionnez la version de Neo4j que vous utilisez.</td>
</tr>
<tr>
<td></td>
<td>Cette liste n’est pas affichée lorsque la case Use an existing connection est cochée.</td>
</tr>
<tr>
<td></td>
<td>Après avoir sélectionné la version de la base de données, il vous est demandé d’installer les fichiers Jar contenant les pilotes correspondant à la base de données.</td>
</tr>
<tr>
<td>Remote server</td>
<td>Cochez cette case si vous utilisez un serveur distant Neo4j et spécifiez l’URL racine dans le champ Server URL.</td>
</tr>
<tr>
<td>Database path</td>
<td>Si vous utilisez Neo4j en mode embarqué, spécifiez le répertoire qui contiendra vos fichiers de données. Le répertoire spécifié sera créé s’il n’existe pas déjà.</td>
</tr>
<tr>
<td></td>
<td>Cette case apparaît uniquement si la case Use an existing connection ou la case Remote Server n’est pas cochée.</td>
</tr>
<tr>
<td>Shutdown after job</td>
<td>Cochez cette case pour arrêter la connexion à la base de données Neo4j lorsqu’aucune opération sur Neo4j n’est effectuée après le composant courant.</td>
</tr>
<tr>
<td></td>
<td>D’une manière alternative, vous pouvez utiliser le tNeo4jClose pour arrêter la base de données.</td>
</tr>
<tr>
<td></td>
<td>Cela permet d’éviter des erreurs comme “Id file not properly shutdown” lors de la prochaine exécution de Jobs impliquant Neo4j.</td>
</tr>
<tr>
<td><strong>Mapping</strong></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>
| Cette case est disponible uniquement lorsque la case *Use an existing connection* est cochée. | Cliquez sur le bouton [...] ou double-cliquez sur le composant dans l’espace de modélisation graphique pour ouvrir l’éditeur de mapping des index et des relations. Utilisez cet éditeur pour indexer un nœud ou créer des relations durant l’insertion d’un nœud.  
- **Auto indexed** : Cochez la case *Auto indexed* pour qu’une colonne indexe automatiquement des nœuds avec cette propriété.  
- **Index creation** : Après avoir sélectionné une colonne, cliquez sur le bouton [+] pour créer sur les nœuds autant d’index que vous le souhaitez, avec la propriété correspondant à la colonne sélectionnée.  
  - **Name** : Spécifiez un nom d’index entre guillemets doubles.  
  - **Key** : Spécifiez une clé d’index entre guillemets doubles.  
  - **Value (empty for current row)** : Spécifiez une valeur d’index entre guillemets doubles. Si vous laissez ce champ vide, la valeur par défaut de l’index ajouté sur chaque nœud deviendra la valeur de la propriété du nœud actuel.  
  - **Unique** : Cochez cette case si vous souhaitez que l’index défini ne soit créé qu’une seule fois dans le graphique plutôt que sur chaque nœud.  
- **Relationship creation** : Après avoir sélectionné une colonne, cliquez sur le bouton [+] pour créer autant de relations que vous le souhaitez pour les nœuds, avec la propriété correspondant à la colonne sélectionnée.  
  - **Type** : Spécifiez un type de relation entre guillemets doubles.  
  - **Direction** : Sélectionnez une direction de relation : *Outgoing* ou *Incoming*.  
  - **Index name** : Spécifiez un nom d’index pour la relation entre guillemets doubles.  
  - **Index key** : Spécifiez une clé d’index pour la relation entre guillemets doubles.  
  - **Value (empty for current row)** : Spécifiez une valeur d’index pour la relation entre guillemets doubles. Si vous laissez ce champ vide, la valeur par défaut de l’index ajouté sur la relation deviendra la valeur de la propriété du nœud actuel. | Cochez cette case afin de créer des nœuds ayant un libellé. Saisissez le nom de votre libellé dans le champ *Label name*.  
Cette case n’est pas affichée si l’option *Neo4J 1.X.X* est sélectionnée dans la liste *DB Version* ou si l’option *Delete* est sélectionnée dans la liste *Data action*. |
| **Data action** | Sur les données du nœud, vous pouvez effectuer les actions suivantes :
| | • **Insert** : ajouter un nouveau nœud à la base de données. 
| | • **Update** : effectuer des modifications sur les entrées existantes. 
| | • **Update or insert** : recherche le nœud via un index pour le mettre à jour ou le modifier. Si le nœud n’existe pas, un nouveau nœud est inséré 
| | • **Delete** : supprime les nœuds récupérés par l’index en fonction du flux d’entrée. 
| **Index name** | Spécifiez le nom de l’index à interroger. Ce champ est disponible uniquement si l’action sélectionnée dans la liste Data action est autre que **Insert**. 
| **Index key** | Spécifiez la clé d’index à interroger. Ce champ est disponible uniquement si l’action sélectionnée dans la liste Data action est autre que **Insert**. 
| **Index value** | Sélectionnez la valeur d’index à interroger. Ce champ est disponible uniquement si l’action sélectionnée dans la liste Data action est autre que **Insert**. 
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs. 
| | Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
| | • **View schema** : sélectionnez cette option afin de voir le schéma. 
| | • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. 
| | • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. 

Notez que cette option fonctionne uniquement à partir de la version 2.0 de Neo4j 2.0 et avec Java 7.
**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Advanced settings

**Commit every**

Saisissez le nombre de lignes à compléter avant de commiter les lots de nœuds dans la base de données. Cette option assure la qualité de la transaction (mais pas le rollback) et permet une meilleure performance lors de l’exécution.

**Avertissement :**

Cette option est supportée uniquement par le mode embarqué de la base de données. Vous ne pouvez pas effectuer de transactions en mode REST.

**Batch import**

Cochez cette case pour activer le mode batch (par lots).

**Avertissement :**

Cette option est supportée uniquement par le mode embarqué de la base de données. Il est recommandé d’effectuer une sauvegarde avant d’exécuter le Job, afin d’éviter la corruption des données.

**Remarque :**

Si vous avez configuré la création d’index sur plusieurs colonnes dans la table Mapping, il est recommandé de cocher la case Unique dans les paramètres d’index pour la dernière colonne afin d’éviter de créer des doublons d’index non désirés susceptibles de causer des problèmes de chargement de batch.


**Node store mapped memory**

Saisissez la taille de la mémoire allouées aux nœuds, en Mo.
### Variables globales

<table>
<thead>
<tr>
<th>Variables globales</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>Nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant est utilisé en tant que composant de sortie et nécessite un lien d’entrée.</td>
<td></td>
</tr>
</tbody>
</table>

---

**Scénario 1 : Écrire des données dans une base de données Neo4j et lire des données spécifiques de cette base de données**

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario simple décrit un Job composé de deux sous-jobs : le premier sous-job lit des données concernant des employés à partir d’un fichier CSV, les écrit dans une base de données Neo4j et déclenche le second sous-job, qui lit les données concernant les employés sous certaines conditions de requête de la base de données Neo4j et les affiche dans la console *Run*. 

---

**Relationship store mapped memory**

<table>
<thead>
<tr>
<th>Description</th>
<th>Saisissez la taille de la mémoire allouée aux relations, en Mo.</th>
</tr>
</thead>
</table>

**Property store mapped memory**

<table>
<thead>
<tr>
<th>Description</th>
<th>Saisissez la taille de la mémoire allouée à la propriété, en Mo.</th>
</tr>
</thead>
</table>

**String store mapped memory**

<table>
<thead>
<tr>
<th>Description</th>
<th>Saisissez la taille de la mémoire allouée aux chaînes de caractères, en Mo.</th>
</tr>
</thead>
</table>

**Array store mapped memory**

<table>
<thead>
<tr>
<th>Description</th>
<th>Saisissez la taille de la mémoire allouée aux tableaux, en Mo.</th>
</tr>
</thead>
</table>

**tStatCatcher Statistics**

<table>
<thead>
<tr>
<th>Description</th>
<th>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.</th>
</tr>
</thead>
</table>
Ajouter et relier les composants

Procédure
1. Créez un Job et ajoutez les composants suivants au Job en saisissant leur nom dans l’espace de modélisation graphique :
   - un composant `tFileInputDelimited`, pour lire les données des employés à partir d’un fichier CSV,
   - un composant `tNeo4jOutput`, pour écrire les données des employés dans une base de données Neo4j,
   - un composant `tNeo4jInput`, pour lire les données des employés à partir de la base de données Neo4j sous des conditions définies et
   - un composant `tLogRow`, pour afficher les données dans la console `Run`.
2. Reliez le composant `tFileInputDelimited` au composant `tNeo4jOutput` à l’aide d’une connexion de type `Row > Main`.
3. Connectez le composant `tNeo4jInput` au composant `tLogRow` à l’aide d’un lien de type `Row > Main`.
4. Reliez le composant `tFileInputDelimited` au composant `tNeo4jInput` à l’aide d’une connexion de type `Trigger > On Subjob Ok`.
5. Renommez les composants afin de mieux identifier leur rôle au sein du Job.

Configurer les composants

Importer des données dans la base de données Neo4j

Procédure
1. Double-cliquez sur le composant `tFileInputDelimited` afin d’ouvrir sa vue `Basic settings`. 
2. Dans le champ **File name/Stream**, spécifiez le chemin d’accès au fichier CSV qui contient les données des employés à lire.

Le fichier d’entrée CSV utilisé dans cet exemple se présente de la manière suivante :

```plaintext
employeeID;employeeName;age;hireDate;salary;managerID
1;Rutherford Roosevelt;38;06-10-2008;13336.58;m5
2;Warren Adams;43;05-22-2008;11626.68;m6
3;Andrew Roosevelt;55;04-01-2007;10052.95;m4
4;Herbert Quincy;54;06-14-2007;10694.71;m6
5;Woodrow Polk;33;08-14-2007;13751.50;m4
6;Theodore Johnson;47;01-26-2008;12426.87;m6
7;Benjamin Adams;32;02-25-2008;10438.65;m4
8;Woodrow Harrison;51;10-11-2008;11188.27;m5
9;George Truman;40;04-28-2008;14254.49;m5
10;Harry Jackson;38;04-01-2008;12798.78;m6
```


4. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir la boîte de dialogue [Schema] et définissez le schéma d’entrée selon la structure du fichier d’entrée. Dans cet exemple, le schéma d’entrée est composé de six colonnes : `employeeID` (Integer), `employeeName` (String), `age` (Integer), `hireDate` (Date), `salary` (Float), et `managerID` (String).

Lorsque vous avez terminé, cliquez sur **OK** pour fermer la boîte de dialogue [Schema] et propager le schéma au composant suivant.
5. Cliquez sur le composant **tNeo4jOutput** et sélectionnez l'onglet **Component** pour ouvrir sa vue **Basic settings**.

6. Définissez une connexion à la base de données Neo4j. Dans cet exemple, la base de données Neo4j est accessible en mode REST. Cochez la case **Remote server** et spécifiez l'URL du serveur Neo4j dans le champ **Server URL**: “http://localhost:7474/db/data” dans cet exemple.

7. Si nécessaire, cliquez sur le bouton **Sync columns** pour vous assurer que le composant ait le même schéma que le composant précédent.

Laissez les autres paramètres tels qu’ils sont.

**Lire des données de la base de données Neo4j**

**Procédure**

1. Double-cliquez sur le composant **tNeo4jInput** pour ouvrir sa vue **Basic settings**.


Les colonnes définies du schéma apparaissent automatiquement dans la table Mapping.
4. Dans le champ Query, renseignez la requête Cypher de sorte qu’elle corresponde aux données à lire de la base de données Neo4j. Dans cet exemple, utilisez la requête Cypher suivante pour trouver les employés qui ont plus de 40 ans et travaillent sous les ordres du manager m6.

```
"MATCH (n) WHERE n.age > 40 AND n.managerID = 'm6' RETURN n;"
```

5. Renseignez le champ Return parameter pour chaque colonne du schéma avec un paramètre de retour entre guillemets doubles afin de mapper les propriétés du nœud dans la base de données Neo4j aux colonnes du schéma.

6. Double-cliquez sur le composant LogRow afin d’ouvrir sa vue Basic settings et sélectionnez l’option Table (print values in cells of a table) afin d’afficher les informations récupérées sous forme de tableau.

Exécuter le Job

Procédure

1. Appuyez sur Ctrl+S pour enregistrer le Job.
2. Appuyez sur F6 ou cliquez sur le bouton Run dans la vue Run pour exécuter le Job.

Les données des employés dans le fichier CSV sont écrites dans la base de données Neo4j. Les informations des employés correspondant aux conditions définies sont récupérées de la base de données Neo4j et affichées dans la console.

Scénario 2 : Ecrire des informations concernant des familles dans Neo4j et créer des relations

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario décrit un Job écrivant des informations relatives à des familles dans des nœuds ayant un label, dans une base de données Neo4j distante et créant des relations basées sur les noms de famille.
Construire le Job

Procédure

1. Créez un Job et ajoutez les composants suivants dans le Job en saisissant leurs noms dans l'espace de modélisation graphique ou en les déposant depuis la Palette :
   - un `tFileInputDelimited` pour lire les données relatives à la famille à partir d'un fichier CSV,
   - un `tNeo4jOutput` pour écrire les données des familles dans une base de données Neo4j et créer des relations entre un mari et une femme.
2. Reliez le composant `tFileInputDelimited` au `tNeo4jOutput` à l'aide d'un lien Row > Main.
3. Renommez les composants afin de mieux identifier leur rôle au sein du Job.

![Diagramme de Job](image)

Configurer les composants

Configurer la source des données

Procédure

1. Double-cliquez sur le composant `tFileInputDelimited` pour ouvrir sa vue Basic settings.

![VUE BASIC SETTINGS](image)

2. Dans le champ File name/Stream, spécifiez le chemin d'accès au fichier CSV contenant les données des familles à lire.
Le fichier CSV d'entrée utilisé dans cet exemple se présente comme suit :

```plaintext
Name;Gender;Age;Family
Jenny;Female;24;the Johnsons
Jack;Male;26;the Johnsons
Richard;Male;35;the Blacks
Anne;Female;36;the Whites
Helen;Female;28;the Blacks
Tom;Male;38;the Whites
```

4. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir la boîte de dialogue [Schema] et définissez le schéma d’entrée, à partir de la structure du fichier d’entrée. Dans cet exemple, le schéma d’entrée se compose de six colonnes: **name** (Integer), **gender** (String), **age** (Integer) et **family** (String).

Cela fait, cliquez sur **OK** pour fermer la boîte de dialogue [Schema] et propagez le schéma au composant suivant.

**Ecrire des données dans Neo4j et créer des index et une relation**

**Procédure**

1. Cliquez sur le composant **tNeo4jOutput** et sélectionnez l’onglet **Component**.
2. Dans la liste DB Version, sélectionnez Neo4j 2.X.X pour activer les libellés des nœuds.
4. Double-cliquez sur le composant tNeo4jOutput ou cliquez sur le bouton [...] du champ Mapping pour ouvrir l’éditeur d’index et de mapping de relations.
5. Sélectionnez la colonne name dans le schéma, cliquez sur l’onglet Index creation, cliquez sur le bouton [+] pour ajouter une ligne à la table, puis créez un index nommé first_name sur cette colonne :
   - Dans le champ Name, saisissez first_name entre guillemets doubles.
   - Dans le champ Key, saisissez first_name entre guillemets doubles pour donner une clé à l’index.
   Cliquez dans le panneau du schéma afin de valider la création de l’index.
6. Sélectionnez la colonne family dans le schéma, cliquez sur l’onglet Index creation puis sur le bouton [+] pour ajouter une ligne à la table. Créez un index nommé family sur cette colonne :
   - Dans le champ Name, saisissez family entre guillemets doubles.
   - Dans le champ Key, saisissez family_name entre guillemets doubles pour donner une clé à l’index.
   Cliquez dans le panneau du schéma afin de valider la création de l’index.
7. Sélectionnez la colonne family dans le schéma, cliquez sur l’onglet Relationship creation, cliquez sur le bouton [+] pour ajouter deux lignes à la table. Créez une relation nommée Spouse sur cette colonne basée sur l’index nommé family :
• Dans le champ **Type**, saisissez *Spouse* entre guillemets doubles.

• Dans la liste **Direction**, sélectionnez *Outgoing* ou *Incoming*.

• Dans le champ **Index Name**, saisissez *family* entre guillemets doubles.

• Dans le champ **Index Key**, saisissez *family_name* entre guillemets doubles.

Cliquez dans le panneau du schéma afin de valider votre création de relation, puis cliquez sur **OK** pour fermer l’éditeur de mapping.

8. Cochez la case **Use label (Neo4j > 2.0)** et saisissez *Families* entre guillemets doubles dans le champ **Label name** afin que les nœuds à créer soient libellés *Families*.

9. Dans la liste **Data action**, sélectionnez *Insert or update* et configurez une clé de référence dans la zone **Index** qui s’affiche :

- Dans le champ **Index name**, saisissez *first_name* entre guillemets doubles.

- Dans le champ **Index key**, saisissez *first_name* entre guillemets doubles.

- Dans le champ **Index value**, sélectionnez *name*. Comme le champ **Value** est vide lors de la création de l’index, la valeur de l’index sera la valeur de la colonne *name* pour chaque ligne.

Lorsque le Job est exécuté, les nœuds sont insérés ou mis à jour dans la base de données Neo4j selon l’index *first_name* : pour chaque ligne de données, si un nœud contenant le même prénom existe déjà dans la base de données, le nœud est mis à jour. Sinon, un nouveau nœud est créé.
Exécuter le Job et vérifier les résultats

Procédure

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder le Job puis appuyez sur **F6** ou cliquez sur **Run** dans la vue **Run** pour exécuter le Job.


   ```cypher
   MATCH (n:`Families`) RETURN n;
   ```

Comme affiché dans le graphique, trois paires de nœuds libellés *Families* sont créés et ceux ayant le même nom de famille sont reliés à l’aide de la relation *Spouse*.
**tNeo4jOutputRelationship**

Ce composant reçoit des données du composant précédent et écrit les relations dans Neo4j.

**Propriétés du tNeo4jOutputRelationship Standard**

Ces propriétés sont utilisées pour configurer le tNeo4jOutputRelationship s’exécutant dans le framework de Jobs Standard.

Le composant tNeo4jOutputRelationship Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Use existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB version</td>
<td>Sélectionnez la version de Neo4j que vous utilisez. Cette liste n’est pas affichée lorsque la case Use an existing connection est cochée. Après avoir sélectionné la version de la base de données, il vous est demandé d’installer les fichiers Jar contenant les pilotes correspondant à la base de données. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
<tr>
<td>Remote server</td>
<td>Cochez cette case si vous utilisez un serveur distant Neo4j et spécifiez l’URL racine dans le champ Server URL.</td>
</tr>
<tr>
<td>Database path</td>
<td>Si vous utilisez Neo4j en mode embarqué, spécifiez le chemin d’accès au fichier de données.</td>
</tr>
<tr>
<td></td>
<td><strong>Avertissement :</strong> Ce champ apparaît uniquement si la case Use an existing connection n’est pas cochée.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
</tbody>
</table>
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In**: Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

**Repository**: Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

**Server url**: Si vous utilisez Neo4j en mode REST, saisissez l’URL racine.

**Avertissement** :

Ce champ apparaît uniquement si la case Remote server est cochée.

**Shutdown after job**: Cochez cette case pour arrêter la connexion à la base de données Neo4j lorsqu’aucune opération sur Neo4j n’est effectuée après le composant courant.

D’une manière alternative, vous pouvez utiliser le tNeo4jClose pour arrêter la base de données.

Cela permet d’éviter des erreurs comme “Id file not properly shutdown” lors de la prochaine exécution de Jobs impliquant Neo4j.

Cette case est disponible uniquement lorsque la case Use an existing connection est cochée.

**Relationship type**: Spécifiez le type de relation à créer pour chaque ligne.

**Start node / Index name**: Spécifiez le nom de l’index pour interroger le nœud de début de la relation la plus récente.

**Start node / Index key**: Spécifiez la clé d’index pour interroger le nœud de début.

**Start node / Index value**: Sélectionnez la valeur de l’index pour interroger le nœud de début.
### Relationship direction

<table>
<thead>
<tr>
<th><strong>Sélectionnez la direction d’une relation.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outgoing</strong> : la relation début du nœud de début et termine au nœud de fin.</td>
</tr>
<tr>
<td><strong>Incoming</strong> : la relation débute du nœud de fin et termine au nœud de début.</td>
</tr>
</tbody>
</table>

### End node / Index name

| Spécifiez le nom de l’index pour interroger le nœud de fin de la relation la plus récente. |

### End node / Index key

| Spécifiez la clé de l’index pour interroger le nœud de fin. |

### End node / Index value

| Sélectionnez la valeur de l’index pour interroger le nœud de fin. |

### Mapping

| Utilisez cette table pour mapper les propriétés de relations et les colonnes du schéma d’entrée. |

### Advanced settings

**Commit every**

<table>
<thead>
<tr>
<th>Saisissez le nombre de lignes à compléter avant de commiter les lots de nœuds dans la base de données. Cette options assure la qualité de la transaction (mais pas le rollback) et permet une meilleure performance lors de l’exécution.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Avertissement</strong> :</td>
</tr>
<tr>
<td>Cette option est supportée uniquement par le mode embarqué de la base de données. Vous ne pouvez pas effectuer de transactions en mode REST.</td>
</tr>
</tbody>
</table>

**Batch import**

<table>
<thead>
<tr>
<th>Cochez cette case pour activer le mode batch (par lots).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Avertissement</strong> :</td>
</tr>
</tbody>
</table>

### Node store mapped memory

| Saisissez la taille de la mémoire allouées aux nœuds, en Mo. |

### Relationship store mapped memory

| Saisissez la taille de la mémoire allouée aux relations, en Mo. |

### Property store mapped memory

| Saisissez la taille de la mémoire allouée à la propriété, en Mo. |

### String store mapped memory

| Saisissez la taille de la mémoire allouée aux chaînes de caractères, en Mo. |
Variables globales

| Variables globales   | NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.
|                     | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaine de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
|                     | Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
|                     | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
|                     | Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

| Règle d’utilisation | Ce composant et utilisé en tant que composant de sortie et nécessite un lien de sortie.

Scénario : Ecrire des informations concernant des acteurs et des films dans Neo4j avec des relations hiérarchiques

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario décrit un Job important des informations concernant des acteurs et des films depuis deux fichiers CSV dans une base de données Neo4j distante et créant une relation entre les acteurs et les films, selon un autre fichier CSV décrivant le rôle des acteurs dans ces films.

Construire le Job

Procédure

1. Créez un Job et ajoutez les composants suivants au Job en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette :
   • un tNeo4jConnection, pour ouvrir une connexion à une base de données Neo4j,
   • trois tFileInputDelimited, pour lire les informations d’entrée concernant les acteurs et les films,
• deux tNeo4jOutput, pour écrire des informations concernant les films et les acteurs dans la base de données Neo4j connectée,

• un tNeo4jOutputRelationship, pour créer une relation entre les acteurs et les films, et

• un tNeo4jClose, pour fermer la connexion précédente à Neo4j après l'exécution du Job.

2. Reliez le premier tFileInputDelimited au premier tNeo4jOutput à l'aide d'un lien Row > Main. Ce sous-job importe les acteurs dans la base de données Neo4j.

3. Connectez le second tFileInputDelimited au second tNeo4jOutput à l'aide d'un lien Row > Main. Ce sous-job importe les films dans la base de données Neo4j.

4. Reliez le troisième tFileInputDelimited au tNeo4jOutputRelationship à l'aide d'un lien Row > Main. Ce sous-job crée une relation entre les acteurs et les films.

5. Reliez les sous-jobs à l'aide de liens Trigger > On Subjob Ok.

6. Renommez les composants afin de mieux identifier leur rôle au sein du Job.
**Configurer les composants**

**Ouvrir une connexion à une base de données Neo4j**

**Procédure**

1. Double-cliquez sur le composant `tNeo4jConnection` pour ouvrir sa vue `Basic settings`.

2. Dans la liste `DB Version`, sélectionnez `Neo4J 2.X.X` pour activer le libellé des nœuds.


**Importer les données des acteurs dans Neo4j**

**Procédure**

1. Double-cliquez sur le premier `tFileInputDelimited` pour ouvrir la vue `Basic settings`.

2. Dans le champ `File name/Stream`, spécifiez le chemin d'accès au fichier CSV décrivant le nom des acteurs, leur année de naissance et leur rôle dans ces films.

   Le fichier CSV d'entrée utilisé dans cet exemple se présente comme suit :

   ```
   Name;Born
   Keanu Reeves;1964
   Laurence Fishburne;1961
   Carrie-Anne Moss;1967
   ```

3. Dans le champ `Header`, spécifiez le nombres de lignes d'en-tête à ignorer. Dans cet exemple, la première ligne du fichier CSV est la ligne d'en-tête, saisissez donc `1`. 


![Schema of Actors](image)

5. Double-cliquez sur le premier tNeo4jOutput pour ouvrir sa vue Component.

![Create 'Actors' (tNeo4jOutput_1)](image)

6. Cochez la case **Use an existing connection** pour réutiliser la connexion à la base de données Neo4j définie par le tNeo4jConnection, seul composant de connexion utilisé dans cet exemple.

7. Double-cliquez sur le tNeo4jOutput ou cliquez sur le bouton Mapping de la vue Basic settings du composant, pour ouvrir l’éditeur d’index et de mapping de relations.

8. Sélectionnez la colonne *name*, dans le schéma, et cliquez sur l’onglet Index creation, puis sur le bouton [+] pour ajouter une ligne à la table. Créez un index nommé *name* dans cette colonne :

   - Dans le nom **Name**, saisissez *name* entre guillemets doubles.

   - Dans le champ **Key**, saisissez *name* entre guillemets doubles pour donner une clé à l’index *name*.

   Cliquez dans le panneau du schéma afin de valider votre création d’index puis cliquez sur OK pour fermer l’éditeur de mapping.
9. Cochez la case **Use label (Neo4j > 2.0)** et saisissez **Actors** entre guillemets doubles dans le champ **Label name**, afin que les nœuds créés soit libellés **Actors**.

**Importer les données des films dans Neo4j**

**Procédure**

1. Double-cliquez sur le second **tFileInputDelimited** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **File name/Stream**, spécifiez le chemin d'accès au fichier CSV contenant les informations des films.

Le fichier CSV d’entrée utilisé dans cet exemple se présente comme suit :

```
Title;Released;Tagline
The Matrix;03-31-1999;Welcome to the Real World
The Matrix Reloaded;05-07-2003;Free Your Mind
```
3. Dans le champ **Header**, spécifiez le nombre de lignes d’en-tête à ignorer. Dans cet exemple, l’en-tête est la première ligne du fichier CSV.


5. Double-cliquez sur le second **tNeo4jOutput** pour ouvrir sa vue **Component**.

![Create 'Movies' (tNeo4jOutput_2)](image)

6. Cochez la case **Use an existing connection** afin de réutiliser la connexion à la base de données Neo4j ouverte par le composant **tNeo4jConnection**, seul composant de connexion utilisé dans cet exemple.

7. Double-cliquez sur le composant **tNeo4jOutput** ou cliquez sur le bouton **Mapping** de la vue **Basic settings** pour ouvrir l’éditeur d’index et de mapping de relations.

8. Sélectionnez la colonne *title* dans le schéma, cliquez sur l’onglet **Index creation**, cliquez sur le bouton [+] pour ajouter une ligne à la table, puis créez un index nommé *title* dans cette colonne :
   - Dans le champ **Name**, saisissez *title* entre guillemets doubles.
   - Dans le champ **Key**, saisissez *title* entre guillemets doubles pour donner une clé à l’index.

Cliquez dans le panneau du schéma pour valider votre création d’index, puis cliquez sur **OK** pour fermer l’éditeur de mapping.
9. Cochez la case **Use label (Neo4j > 2.0)** et saisissez **Movies** entre guillemets doubles dans le champ **Label name**, pour que les nœuds créés aient le libellé **Movies**.

**Créer la relation acteur-film dans Neo4j**

**Procédure**

1. Double-cliquez sur le troisième **tFileInputDelimited** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **File name/Stream**, spécifiez le chemin d'accès au fichier CSV décrivant le rôle des acteurs dans les films.

Le fichier d’entrée CSV utilisé dans cet exemple se présente comme suit :

```
Actor;Movie;Role
Keanu Reeves;The Matrix;Neo
Keanu Reeves;The Matrix Reloaded;Neo
```


5. Double-cliquez sur le **tNeo4jOutputRelationship** pour ouvrir sa vue **Basic settings**.

6. Cochez la case **Use an existing connection** pour réutiliser la connexion à la base de données Neo4j ouverte par le composant **tNeo4jConnection**, seul composant de connexion utilisé dans cet exemple.

7. Dans le champ **Relationship type**, saisissez le type de relation à créer, entre guillemets doubles. Dans cet exemple, le type de relation est *ACTED_IN* (a joué dans), qui apparaît comme libellé des relations entre les nœuds.

8. Définissez les nœuds d’entrée et de sortie, ainsi que la direction de la relation à créer. Dans cet exemple, la relation est créée à partir du nœud *Actors*, sur lequel un index nommé *name* a été défini, vers le nœud *Movies*, sur lequel un index nommé *title* a été défini.

   - **Start node** : saisissez *name* entre guillemets doubles dans les champs **Index name** et **Index key**, puis sélectionnez la colonne *name* du schéma, dans la liste **Index value**. Comme le champ **Value** est laissé vide durant la création de l’index, la valeur de l’index est la valeur de la colonne *name* pour chaque ligne.
Dans la liste **Relationship direction**, sélectionnez **Outgoing**.

- **End node** : saisissez *title* entre guillemets doubles dans les champs **Index name** et **Index key**, puis sélectionnez la colonne *title* du schéma, dans la liste **Index value**. Comme le champ **Value** est laissé vide durant la création de l’index, la valeur de l’index est la valeur de la colonne *title* pour chaque ligne.

9. Cliquez trois fois sur le bouton [+ ] pour ajouter trois lignes à la table **Mapping**, afin de définir les trois propriétés de la relation :

| Propriété | Valeur récupérée de...
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>La colonne <em>actor</em> du schéma</td>
</tr>
<tr>
<td>Movie</td>
<td>La colonne <em>movie</em> du schéma</td>
</tr>
<tr>
<td>Role</td>
<td>La colonne <em>role</em> du schéma</td>
</tr>
</tbody>
</table>

**Exécuter le Job et vérifier les résultats**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job et appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour l’exécuter.


   ```cypher
   MATCH (a)-[:`ACTED_IN`]->(b) RETURN a,b;
   ```

Comme vous pouvez le voir dans le graphique, les nœuds *Actors* sont reliés aux nœuds *Movies* par la relation *ACTED_IN*, avec les propriétés définies dans le Job.
**tNeo4jRow**

Ce composant exécute les requêtes Cypher sur la base de données Neo4j spécifiée.

Selon la nature de la requête, le tNeo4jRow agit sur la structure même de la base de données ou sur les données (mais sans les manipuler).

Le composant tNeo4jRow est le composant spécifique pour les requêtes, pour cette base de données. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

**Propriétés du tNeo4jRow Standard**

Ces propriétés sont utilisées pour configurer le tNeo4jRow s’exécutant dans le framework de Jobs Standard.

Le composant tNeo4jRow Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>DB version</strong></td>
<td>Sélectionnez la version de Neo4j que vous utilisez. Cette liste n’est pas affichée lorsque la case Use an existing connection est cochée. Après avoir sélectionné la version de la base de données, il vous est demandé d’installer les fichiers Jar contenant les pilotes correspondant à la base de données. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
<tr>
<td><strong>Remote server</strong></td>
<td>Cochez cette case si vous utilisez un serveur distant Neo4j et spécifiez l’URL racine dans le champ Server URL.</td>
</tr>
<tr>
<td><strong>Database path</strong></td>
<td>Si vous utilisez Neo4j en mode embarqué, spécifiez le répertoire qui contiendra vos fichiers de données. Le répertoire spécifié sera créé s’il n’existe pas déjà. Cette case apparaît uniquement si la case Use an existing connection ou la case Remote Server n’est pas cochée.</td>
</tr>
<tr>
<td><strong>Shutdown after job</strong></td>
<td>Cochez cette case pour arrêter la connexion à la base de données Neo4j lorsqu’aucune opération sur Neo4j n’est effectuée après le composant courant. D’une manière alternative, vous pouvez utiliser le tNeo4jClose pour arrêter la base de données.</td>
</tr>
</tbody>
</table>
Cela permet d'éviter des erreurs comme "Id file not properly shutdown" lors de la prochaine exécution de Jobs impliquant Neo4j.

Cette case est disponible uniquement lorsque la case **Use an existing connection** est cochée.

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réserve **line** lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center ([https://help.talend.com](https://help.talend.com)).

**Query**

Saisissez votre requête Cypher entre guillemets doubles. Si vous avez des paramètres correspondant aux colonnes du schéma d’entrée, déclarez-les entre accolades (**) et mappez-les dans la table **Parameters**.

**Parameters**

Cliquez sur le bouton [+*] pour ajouter des paramètres correspondant à votre requête Cypher et mappez-les avec les colonnes du schéma d’entrée.

- **Parameter name** : Saisissez le nom de votre paramètre entre guillemets doubles.
<table>
<thead>
<tr>
<th><strong>Parameter value</strong></th>
<th>Sélectionnez la colonne de schéma avec laquelle vous souhaitez mapper votre paramètre.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Die on error</strong></td>
<td>Cette case est cochée par défaut. Décochez-la pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>tStatCatcher Statistics</strong></th>
<th>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.</th>
</tr>
</thead>
</table>
| **Commit every**            | Saisissez le nombre de lignes à compléter avant de committer les lots de nœuds dans la base de données. Cette options assure la qualité de la transaction (mais pas le rollback) et permet une meilleure performance lors de l’exécution.  
Cette option est supportée uniquement par le mode embarqué de la base de données. Vous ne pouvez pas effectuer de transactions en mode REST, c’est-à-dire pas sur un serveur distant dans les versions dépréciées 2.X.X. |

**Variables globales**

<table>
<thead>
<tr>
<th><strong>Variables globales</strong></th>
<th><strong>NB_NODE_INSERTED</strong></th>
<th>nombre de nœuds insérés. Cette variable est une variable <em>After</em> et retourne un long.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>NB_RELATIONSHIP_INSERTED</strong></td>
<td>nombre de relations insérées. Cette variable est une variable <em>After</em> et retourne un long.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_PROPERTY_UPDATED</strong></td>
<td>nombre de propriétés mises à jour. Cette variable est une variable <em>After</em> et retourne un long.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_NODE_DELETED</strong></td>
<td>nombre de nœuds supprimés. Cette variable est une variable <em>After</em> et retourne un long.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_RELATIONSHIP_DELETED</strong></td>
<td>nombre de relations supprimées. Cette variable est une variable <em>After</em> et retourne un long.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Utilisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Règle d'utilisation</td>
</tr>
</tbody>
</table>

**Scénario 1 : Créer des nœuds avec un libellé à l'aide d'une requête Cypher**

Ce scénario s'applique uniquement aux solutions Talend avec Big Data.

Ce scénario décrit un Job qui, à l'aide d'une requête Cypher, crée des nœuds avec un libellé dans une base de données Neo4j embarquée, puis affiche quelques propriétés du nœud défini dans la console.

**Ajouter et relier les composants**

**Procédure**

1. Créez un Job et ajoutez les composants suivants au Job en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant de la Palette :
   - un composant `tNeo4jConnection`, pour ouvrir une connexion à une base de données Neo4j,
   - un composant `tNeo4jRow`, pour créer des nœuds dans la base de données Neo4j à l'aide d'une requête Cypher,
   - un composant `tNeo4jIntput`, pour lire les propriétés définies des nœuds créés,
   - un composant `tLogRow`, pour afficher les données dans la console Run et
   - un composant `tNeo4jClose`, pour fermer la connexion à la base de données Neo4j établie par le composant `tNeo4jConnection`.

2. Reliez le composant `tNeo4jConnection` au composant `tNeo4jRow` à l'aide d'une connexion de type Trigger > On Subjob Ok.

3. Reliez le composant `tNeo4jRow` au composant `tNeo4jIntput` à l'aide d'une connexion de type Trigger > On Subjob Ok.

4. Reliez le composant `tNeo4jIntput` au composant `tLogRow` à l'aide d'une connexion de type Row > Main.

5. Reliez le composant `tNeo4jIntput` au composant `tNeo4jClose` à l'aide d'une connexion de type Trigger > On Subjob Ok.
Configurer les composants

Configurer une connexion à une base de données Neo4j et la création de nœuds

Procédure

1. Double-cliquez sur le composant tNeo4jConnection pour ouvrir sa vue Basic settings dans la vue Component.

2. Dans la liste DB Version, sélectionnez Neo4j 2.X.X pour activer le support des libellés des nœuds.

3. Dans le champ Database path, spécifiez le répertoire pour vos fichiers de données, "E:/Talend/DB/Neo4j/data" dans cet exemple.

Dans cet exemple, utilisez Neo4j en mode embarqué. Pour vous connecter à un serveur Neo4j distant via REST, cochez la case Use a remote server et spécifiez l’URL du serveur Neo4j.
4. Double-cliquez sur le composant **tNeo4jRow** pour ouvrir sa vue **Basic settings** dans la vue **Component**.

5. Cochez la case **Use an existing connection** pour réutiliser la connexion à la base de données Neo4j ouverte par le composant **tNeo4jConnection**, qui est le seul composant de connexion utilisé dans cet exemple.

6. Dans le champ **Query**, renseignez la requête Cypher que le composant doit exécuter.
Dans cet exemple, saisissez la requête suivante pour créer deux nœuds avec pour libellé **Managers** et trois propriétés - *id, name et position* :

```
"CREATE (a:Managers { id: 1, name: 'Gerald White', position: 'HR directory'})
CREATE (b:Managers { id: 2, name: 'Jimmy Black', position: 'Sales manager'})"
```

Étant donné que ce composant n’a pas de flux d’entrée, laissez le schéma et les paramètres vides.

**Configurer la récupération et l’affichage des données**

**Procédure**

1. Double-cliquez sur le composant **tNeo4jInput** pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Use an existing connection** pour réutiliser la connexion ouverte par le composant **tNeo4jConnection**.
3. Cliquez sur le bouton [...] à côté de Edit schema et définissez le schéma correspondant aux propriétés du nœud que vous souhaitez récupérer et afficher. Dans cet exemple, le schéma est composé de deux colonnes, name et position, toutes les deux de type String. Lorsque vous avez terminé, cliquez sur OK pour fermer la boîte de dialogue [Schema] et propager le schéma au composant suivant.

Les colonnes définies du schéma apparaissent automatiquement dans la table Mapping.

4. Dans le champ Query, saisissez la requête Cypher de sorte qu'elle corresponde aux données à lire de la base de données Neo4j. Dans cet exemple, utilisez la requête Cypher suivante pour trouver le nom et la position du manager dont l'ID est 2.

"MATCH (ee:Managers{id:2}) RETURN ee.name, ee.position;"

ou

"MATCH (ee:Managers) WHERE ee.id = 2 RETURN ee.name, ee.position;"

5. Renseignez le champ Return parameter pour chaque colonne de schéma avec un paramètre de retour entre guillemets doubles afin de mapper les propriétés du nœud dans la base de données Neo4j avec les colonnes du schéma.

6. Double-cliquez sur le composant tLogRow afin d'ouvrir sa vue Basic settings et sélectionnez l'option Table (print values in cells of a table) pour afficher les informations récupérées dans la table.

**Exécuter le Job**

**Procédure**

1. Appuyez sur Ctrl+S pour enregistrer le Job.
2. Appuyez sur F6 ou cliquez sur Run dans la vue Run pour exécuter le Job.
Deux nœuds sont créés dans la base de données Neo4j d’après votre requête Cypher et les fichiers de données sont écrits dans le répertoire spécifié. En outre, les informations définies sont récupérées et affichées dans la console.

Scénario 2 : Importer des données dans une base de données Neo4j à partir d’un fichier CSV à l’aide d’une requête Cypher

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario décrit un Job qui importe des données concernant des employés dans une base de données Neo4j à partir d’un fichier CSV, à l’aide d’une requête Cypher, puis affiche les informations dans la console.

Ajouter et lier les composants

Procédure

1. Créez un Job et ajoutez les composants suivants au Job en saisissant leur nom dans l’espace graphique de modélisation ou en les déposant de la Palette :
   - un composant tNeo4jConnection, pour ouvrir une connexion à une base de données Neo4j,
   - un composant tFileInputDelimited, pour lire les données source d’un fichier CSV,
   - un composant tNeo4jRow, pour écrire les données des employés dans la base de données Neo4j à l’aide d’une requête Cypher,
   - un composant tNeo4jInput, pour lire les données des employés de la base de données Neo4j,
   - un composant tLogRow, pour afficher les données dans la console Run et
   - un composant tNeo4jClose, pour fermer la connexion à la base de données Neo4j ouverte par le composant tNeo4jConnection.
2. Reliez le composant tNeo4jConnection au composant tFileInputDelimited à l’aide d’une connexion de type Trigger > On Subjob Ok.
3. Reliez le composant tFileInputDelimited au composant tNeo4jRow à l’aide d’une connexion de type Row > Main.
4. Reliez le composant tFileInputDelimited au composant tNeo4jInput à l’aide d’une connexion de type Trigger > On Subjob Ok.
5. Reliez le composant tNeo4jInput au composant tLogRow à l’aide d’une connexion de type Row > Main.
6. Reliez le composant `tNeo4jInput` au composant `tNeo4jClose` à l’aide d’une connexion de type Trigger > On Subjob Ok.

7. Nommez les composants afin de mieux identifier leur rôle dans le Job.

**Configurer les composants**

**Configurer une connexion à une base de données Neo4j**

**Procédure**

1. Double-cliquez sur le composant `tNeo4jConnection` pour ouvrir sa vue `Basic settings` dans la vue `Component`.

2. Dans la liste `DB Version`, sélectionnez Neo4J 2.X.X.

Dans cet exemple, vous utilisez Neo4j en mode REST. Pour vous connecter à un serveur Neo4j distant en mode embarqué, décochez la case **Use a remote server** et spécifiez le répertoire des données Neo4j dans le champ **Database path**.

**Configurer l’import de données**

**Procédure**

1. Double-cliquez sur le composant **tFileInputDelimited** afin d’ouvrir sa vue **Basic settings** dans la vue **Component**.

2. Dans le champ **File name/Stream**, spécifiez le chemin d'accès au fichier CSV qui contient les données sur les employés à lire.

Le fichier d’entrée CSV utilisé dans cet exemple se présente de la manière suivante :

```
employeeID;employeeName;age;hireDate;salary;managerID
1;Rutherford Roosevelt;38;06-10-2008;13336.58;m5
2;Warren Adams;43;05-22-2008;11626.68;m6
3;Andrew Roosevelt;55;04-01-2007;10052.95;m4
4;Herbert Quincy;54;06-14-2007;10694.71;m6
5;Woodrow Polk;33;08-14-2007;13751.50;m4
6;Theodore Johnson;47;01-26-2008;12426.87;m6
7;Benjamin Adams;32;02-25-2008;10438.65;m4
8;Woodrow Harrison;51;10-11-2008;11188.27;m5
9;George Truman;40;04-28-2008;14254.49;m5
10;Harry Jackson;38;04-01-2008;12798.78;m6
```

3. Dans le champ **Header**, spécifiez le nombre de lignes considérées comme étant des lignes d'en-tête à sauter. Dans cet exemple, la première ligne du fichier CSV est la ligne d'en-tête.

4. Cliquez sur le bouton [...] à côté de **Edit schema** pour ouvrir la boîte de dialogue [Schema] et définissez le schéma d’entrée selon la structure du fichier d’entrée. Dans cet exemple, le schéma d’entrée est composé de six colonnes : **employeeID** (Entier), **employeeName** (Chaîne de caractères), **age** (Entier), **hireDate** (Date), **salary** (Double), et **managerID** (Chaîne de caractères).

Lorsque vous avez terminé, cliquez sur **OK** pour fermer la boîte de dialogue [Schema] et propage le schéma au composant suivant.
5. Double-cliquez sur le composant **tNeo4jRow** pour ouvrir sa vue **Basic settings** dans la vue **Component**.

6. Cochez la case **Use an existing connection** pour réutiliser la connexion à la base de données Neo4j ouverte par le composant **tNeo4jConnection**, qui est le seul composant de connexion utilisé dans cet exemple.

7. Dans le champ **Query**, saisissez la requête Cypher que le composant doit exécuter.
   Dans cet exemple, renseignez les requêtes suivantes pour créer des nœuds avec le nom **Employees** et six propriétés afin de contenir les données du flux d'entrée :
   - **ID**, qui prend la valeur de la variable *id*,
   - **Name**, qui prend la valeur de la variable *name*,
• *Age*, qui prend la valeur de la variable *age*,
• *HireDate*, qui prend la valeur de la variable *hire_date*,
• *Salary*, qui prend la valeur de la variable *salary* et
• *ManagerID*, qui prend la valeur de la variable *manager_id*.

```
"CREATE (n:Employees{ID:{id}, Name:{name}, Age:{age}, HireDate:{hire_date}, Salary:{salary}, ManagerID:{manager_id}})"
```

8. Dans la table *Parameters*, renseignez les variables dans le champ *Parameter*, conformément à votre requête Cypher et mappez chacune d’entre elle avec une colonne du schéma d’entrée en la sélectionnant dans la liste de champs *Parameter value*.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Parameter name</th>
<th>Parameter value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>employeesID</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>employeesName</td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>age</td>
<td></td>
</tr>
<tr>
<td>hire_date</td>
<td>hireDate</td>
<td></td>
</tr>
<tr>
<td>salary</td>
<td>salary</td>
<td></td>
</tr>
<tr>
<td>manager_id</td>
<td>managerID</td>
<td></td>
</tr>
</tbody>
</table>

**Configurer la récupération et l’affichage des données**

**Procédure**

1. Double-cliquez sur le composant *tNeo4jInput* pour ouvrir sa vue *Basic settings*.

2. Cochez la case *Use an existing connection* pour réutiliser la connexion ouverte par le composant *tNeo4jConnection*.
3. Cliquez sur le bouton [...] à côté de *Edit schema* et définissez le schéma correspondant aux propriétés du nœud que vous souhaitez récupérer et afficher.
Lorsque vous avez terminé, cliquez sur OK pour fermer la boîte de dialogue [Schema] et propager le schéma au composant suivant.

Les colonnes définies du schéma apparaissent automatiquement dans la table Mapping.

4. Dans le champ Query, saisissez la requête Cypher de sorte qu'elle corresponde aux données à lire de la base de données Neo4j. Dans cet exemple, utilisez la requête Cypher suivante pour récupérer toutes les données de tous les nœuds avec le nom Employees.

```
“MATCH (n:Employees) RETURN *;”
```

5. Renseignez le champ Return parameter pour chaque colonne de schéma avec un paramètre de retour entre guillemets doubles afin de mapper les propriétés du nœud dans la base de données Neo4j avec les colonnes de schéma.

6. Double-cliquez sur le composant tLogRow afin d'ouvrir sa vue Basic settings et sélectionnez l'option Table (print values in cells of a table) pour afficher les informations récupérées dans une table.

**Exécuter le Job**

**Procédure**

1. Appuyez sur Ctrl+S pour enregistrer le Job.
2. Appuyez sur F6 ou cliquez Run dans la vue Run pour exécuter votre Job.
Les données concernant les employés dans le fichier CSV sont écrites dans la base de données Neo4j et affichées dans la console.

Scénario 3 : Importer des données d'un fichier CSV dans Neo4j et créer des relations à l'aide d'une requête Cypher

Ce scénario s'applique uniquement aux solutions Talend avec Big Data.

Ce scénario décrit un Job important des informations relatives à des familles depuis un fichier CSV dans une base de données Neo4j distante et créant des relations entre les personnes et les familles à l'aide d'une seule requête Cypher via un composant tNeo4jRow.

Le fichier CSV duquel importer les données se présente comme suit dans cet exemple :

<table>
<thead>
<tr>
<th>Name;Gender;Age;Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jenny;Female;24;the Johnsons</td>
</tr>
<tr>
<td>Jack;Male;26;the Johnsons</td>
</tr>
<tr>
<td>Richard;Male;35;the Blacks</td>
</tr>
<tr>
<td>Anne;Female;36;the Whites</td>
</tr>
<tr>
<td>Helen;Female;28;the Blacks</td>
</tr>
<tr>
<td>Tom;Male;38;the Whites</td>
</tr>
</tbody>
</table>

Comme une opération `MERGE` est utilisée avec `LOAD CSV` dans cet exemple, pour assurer que la requête Cypher est exécutée efficacement, utilisez un autre tNeo4jRow pour créer un index sur la propriété à fusionner.

Construire le Job

Procédure

1. Créez un Job et ajoutez deux composants tNeo4jRow au Job en saisissant le nom du composant dans l'espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez les composants à l'aide d'un lien Trigger > OnSubjobOk.
3. Renommez les composants afin de mieux identifier leur rôle au sein du Job.
Configurer les composants

Créer un index

Procédure

1. Double-cliquez sur le premier tNeo4jRow pour ouvrir sa vue Basic settings.

2. Dans la liste DB Version, sélectionnez Neo4J 2.X.X.


4. Dans le champ Query, saisissez la requête suivante pour créer un index sur la propriété à fusionner, name des nœuds Family dans cet exemple : "CREATE INDEX ON :Family(name)"

Importer des données et créer des relations

Procédure

1. Double-cliquez sur le second tNeo4jRow pour ouvrir sa vue Basic settings.
2. Dans la liste DB Version, sélectionnez Neo4J 2.X.X.


4. Dans le champ Query, saisissez la requête Cypher suivante pour importer les données de la famille depuis le fichier CSV, créez les nœuds Person et Family correspondants et créez les relations entre les personnes et les familles :

```
"LOAD CSV WITH HEADERS FROM 'file:E:/Talend/Data/Input/families.csv' AS csvLine
FIELDTERMINATOR ','
MERGE (family:Family { name: csvLine.Family })
CREATE (person:Person { name: csvLine.Name, gender: csvLine.Gender, age: toInt(csvLine.Age)})
CREATE (person)-[:From]->(family)"
```

Exécuter le Job et vérifier le résultat

Procédure

1. Appuyez sur les touches Ctrl+S afin de sauvegarder le Job et appuyez sur F6 ou cliquez sur le bouton Run de la vue Run pour l’exécuter.

2. Dans la barre d’adresse de votre navigateur Web, saisissez l’URL de la base de données Neo4j, http://localhost:7474/ dans cet exemple et saisissez la requête Cypher suivante dans une invite de commande afin de voir les nœuds Person et Family reliés par la relation From :

```
MATCH (a:Person)-[:From]->(b:Family) RETURN a,b;
```
Comme vous pouvez le constater dans le graphique, les nœuds ayant pour libellé *Family* et *Person* ont été créés et les nœuds des personnes de la même famille sont liés au nœud *Family* correspondant via la relation *From*.
tNetezzaBulkExec

Ce composant améliore les performances lors d’opérations Insert dans une base de données Netezza.

**Propriétés du tNetezzaBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tNetezzaBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tNetezzaBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux.
<table>
<thead>
<tr>
<th><strong>Nom</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur de base de données.  
|                  | Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Table            | Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit exister pour que l’opération d’Insert soit autorisée.  
|                  | **Schema et Edit Schema**  
|                  | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
|                  | Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
|                  | • View schema : sélectionnez cette option afin de voir le schéma.  
|                  | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
|                  | • Update repository connection : sélectionnez cette option afin de modiﬁer le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modiﬁcations uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content.  
| File Name        | Nom du fichier à charger. Ce fichier doit se situer sur la même machine que le Studio Talend ou le Job contenant un tNetezzaBulkExec, afin que le composant fonctionne correctement. |
## Advanced settings

<table>
<thead>
<tr>
<th>Field Separator</th>
<th>Caractère, chaîne ou expression régulière séparant les champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Escape character</td>
<td>Caractère d’échappement de la ligne.</td>
</tr>
</tbody>
</table>
| Date format / Date delimiter | **Date format**: Utilisez ce champ pour représenter l’ordre des mois, des jours et des années sous forme de chaîne de caractères.  
**Date delimiter**: Utilisez ce champ pour définir l’élément séparateur entre les valeurs. |
| Time format/ Time delimiter | **Time format**: Utilisez ce champ pour représenter le format de l’heure sous forme de chaîne de caractères.  
**Time delimiter**: Utilisez ce champ pour définir l’élément séparateur entre les valeurs. |
| Encoding | Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données. |
| Max Errors | Saisissez le nombre maximal d’erreurs possibles avant que le processus ne s’arrête. |
| Skip Rows | Renseignez le nombre de lignes à ignorer. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

## Global Variables

| Global Variables | **NB_LINE**: nombre de lignes traitées. Cette variable est une variable After et retourne un entier.  
**ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Variables</td>
<td></td>
</tr>
</tbody>
</table>
### Utilisation

| Règle d'utilisation | Ce composant est principalement utilisé lorsque des transformations non particulières sont requises sur les données à charger dans la base de données Netezza. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue **Basic settings** devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**. |
| Limitation | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com). |

### Scénarios associés

Pour des scénarios associés au composant **tNetezzaBulkExec**, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du tMysqlOutputBulk**.
- **Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec**.
• Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du tOracleBulkExec.
tNetezzaClose

Ce composant ferme une connexion à la base de données Netezza.

**Propriétés du tNetezzaClose Standard**

Ces propriétés sont utilisées pour configurer le tNetezzaClose s’exécutant dans le framework de Jobs Standard.

Le composant tNetezzaClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant <strong>tNetezzaConnection</strong> dans la liste.</td>
</tr>
</tbody>
</table>

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tNetezzaCommit

Ce composant valide les données traitées à travers le Job dans la base de données Netezza connectée.
Le tNetezzaCommit utilise une connexion unique pour commiter en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet donc un gain de performance.

Propriétés du tNetezzaCommit Standard

Ces propriétés sont utilisées pour configurer le tNetezzaCommit s’exécutant dans le framework de Jobs Standard.
Le composant tNetezzaCommit Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant <strong>tNetezzaConnection</strong> dans la liste.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décrochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
<tr>
<td></td>
<td><strong>Avertissement :</strong></td>
</tr>
<tr>
<td></td>
<td>Si vous utilisez un lien de type <strong>Row &gt; Main</strong> pour relier le <strong>tNetezzaCommit</strong> à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case <strong>Close connection</strong> car la connexion sera fermée avant la fin du commit de votre première ligne.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics                                               | Cochez cette case pour collecter les données de log au niveau du composant. |
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants Netezza et notamment avec les composants tNetezzaConnection et tNetezzaRollback.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Scénario associé

Pour plus d’informations relatives au fonctionnement du composant tNetezzaCommit, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tNetezzaConnection

Ce composant ouvre une connexion à une base de données pouvant être réutilisée dans le ou les sous-job(s) suivant(s).

Propriétés du tNetezzaConnection Standard

Ces propriétés sont utilisées pour configurer le tNetezzaConnection s’exécutant dans le framework de Jobs Standard.

Le composant tNetezzaConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Table Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Additional JDBC Parameters</td>
<td>Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.</td>
</tr>
</tbody>
</table>
### Use or register a shared DB Connection

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.

### Advanced settings

#### Auto Commit

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

#### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu’au niveau de chaque composant.

### Utilisation

#### Règle d’utilisation

Ce composant est généralement utilisé avec des composants Netezza, notamment les composants tNetezzaCommit et tNetezzaRollback.

#### Limitation

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez
Scénario associé

Pour un scénario associé au composant tNetezzaConnection, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tNetezzaInput

Ce composant lit une base de données Netezza et en extrait des champs à l’aide d’une requête.

Le tNetezzaInput exécute une requête de base de données dans laquelle l’ordre doit strictement correspondre à l’ordre défini dans le schéma. Les champs sont transmis au composant via une connexion Main row.

Propriétés du tNetezzaInput Standard

Ces propriétés sont utilisées pour configurer le tNetezzaInput s’exécutant dans le framework de Jobs Standard.

Le composant tNetezzaInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionne le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Remarque : Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une</td>
</tr>
</tbody>
</table>
connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par
le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion
tà la base de données à partager, dans
la vue **Basic settings** du composant de
connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant
dédié à la création de connexion, afin de
lire la connexion enregistrée.

Pour plus d’informations concernant le partage
d’une connexion à travers différents niveaux
de Jobs, consultez le **Guide utilisateur du Studio
Talend**.

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. **Built-in** : Le schéma sera créé et conservé pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**. **Repository** : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez... |
propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th>Table name</th>
<th>Nom de la table de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query type et Query</td>
<td>Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs correspond à celui défini dans le schéma.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Use cursor</th>
<th>Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d’optimiser les performances.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim all the String/Char columns</td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td>Trim column</td>
<td>Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| Global Variables | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. 
**QUERY** : requête traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.
**ERROR_MESSAGE** : message d’erreur générée par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| Règle d’utilisation | Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Netezza. |
| Dynamic settings | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

| Limitation | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com). |

### Scénarios associés

Consultez les scénarios du composant tNetezzaInput dans :

- **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520.
tNetezzaNzLoad

Ce composant insère des données dans la table d’une base de données Netezza via l’utilitaire nzload de Netezza.

Le tNetezzaNzLoad charge en masse des données dans une table Netezza, soit à partir d’un fichier, soit à partir d’un flux d’entrée, soit à partir d’un tube nommé.

Propriétés du tNetezzaNzLoad Standard

Ces propriétés sont utilisées pour configurer le tNetezzaNzLoad s’exécutant dans le framework de Jobs Standard.

Le composant tNetezzaNzLoad Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table dans laquelle les données doivent être insérées.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td>None : n’effectuer aucune opération de table.</td>
</tr>
<tr>
<td></td>
<td>Drop and create the table : supprimer la table puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td>Create a table : créer une table qui n’existe pas encore.</td>
</tr>
<tr>
<td></td>
<td>Create table if doesn’t exist : créer la table si nécessaire.</td>
</tr>
<tr>
<td></td>
<td>Clear a table : supprimer le contenu de la table.</td>
</tr>
<tr>
<td><strong>Truncate table</strong></td>
<td>supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <code>line</code> lors du nommage des champs.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Data file</strong></td>
<td>Chemin d'accès complet au fichier de données à utiliser. Si ce composant est utilisé seul (non connecté à un autre composant par un flux d'entrée), alors saisissez le nom d'un fichier existant à charger dans la base de données. S'il est connecté à un autre composant par un flux d'entrée, saisissez le nom du fichier à générer et à écrire avec les données d'entrée, afin de l'utiliser plus tard avec nzload pour charger dans la base de données.</td>
</tr>
<tr>
<td><strong>Use named-pipe</strong></td>
<td>Cochez cette case afin d'utiliser un tube nommé à la place d'un fichier de données. Cette option ne peut être utilisée que lorsque le composant est connecté à un autre composant par un flux d'entrée. Quand la case est cochée, aucun fichier de données n'est généré, et les données sont transférées à l'utilitaire nzload via un tube nommé. Cette option améliore grandement les performances sous Linux et Windows.</td>
</tr>
</tbody>
</table>

**Remarque :**
Ce composant, en mode tube nommé, utilise une interface JNI pour créer un tube nommé et écrire dedans, dans toute plateforme Windows.
Le chemin d'accès associé au fichier DLL JNI
doit être configuré dans le chemin d’accès à la bibliothèque Java. Le composant est automatiquement fourni dans le Studio Talend avec deux DLL pour les systèmes d’exploitation de 32 et 64 bits.

| Named-pipe name | Spécifiez un nom pour le tube nommé à utiliser. Vérifiez que le nom saisi est valide. |

**Advanced settings**

| Use existing control file | Cochez cette case pour fournir un fichier de contrôle à utiliser avec l’utilitaire nzload au lieu de spécifier explicitement toutes les options dans le composant. Lorsque cette case est cochée, l’option *Data file* ainsi que les autres options liées au nzload ne s’appliquent plus. Veuillez vous référer au manuel du nzload de Netezza pour plus d’informations concernant la création d’un fichier de contrôle.  

**Remarque :** La variable globale `NB_LINE` n’est pas supportée lors de l’utilisation d’un fichier de contrôle. |

| Control file | Saisissez le chemin d’accès au fichier de contrôle à utiliser, entre guillemets doubles, ou cliquez sur le bouton [...] afin de parcourir votre répertoire jusqu’au fichier de contrôle. Cette option est passée à l’utilitaire nzload via l’argument `-cf`. |

| Field separator | Caractère, chaîne de caractères ou expression régulière utilisé pour séparer les champs.  

**Avertissement :** Argument `delim` de l’utilitaire nzload. Si vous n’utilisez pas l’option *Wrap quotes around fields*, vous devez être sûr que le séparateur n’est pas inclus dans les données insérées dans la base de données. La valeur par défaut est `\t` ou `TAB`. Pour améliorer les performances, utilisez la valeur par défaut. |

| Wrap quotes around fields | Cette option ne s’applique qu’aux colonnes de type `String`, `Byte`, `Byte[]`, `Char`, et `Object`. Sélectionnez soit :  

**None :** n’entoure pas les valeurs des colonnes de guillemets.  

**Single quote :** entoure les valeurs des colonnes de guillemets simples.  

**Double quote :** entoure les valeurs des colonnes de guillemets doubles.  

**Avertissement :** Si vous utilisez l’option *Single quote* ou *Double quote*, il est nécessaire d’utiliser `\` en tant que `Escape char`. |
### Advanced options
Définissez les arguments nzload dans le tableau correspondant. Cliquez sur le bouton `[*]` autant de fois que nécessaire afin d’ajouter les arguments au tableau.

Cliquez sur le champ **Parameter** et choisissez l’argument souhaité dans la liste, puis cliquez sur le champ **Value** correspondant et saisissez une valeur entre guillemets.

Pour plus d’informations concernant les paramètres disponibles, consultez **Paramètres** à la page 2849.

### Encoding
Sélectionnez l’encodage à partir de la liste.

### Specify nzload path
Cochez cette case afin de spécifier le chemin d’accès complet à l’exécutable nzload. Vous devez cocher cette case si le chemin d’accès n’est pas spécifié dans la variable d’environnement PATH.

### Full path to nzload executable
Chemın d’accès complet à l’exécutable nzload sur la machine en cours d’utilisation. Il est recommandé de spécifier le chemin d’accès dans la variable d’environnement PATH au lieu de sélectionner cette option.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Variables globales

#### Global Variables

- **NB_LINE** : nombre de lignes traitées. Cette variable est une variable *After* et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

#### Règle d’utilisation
Ce composant est principalement utilisé lorsque des transformations particulières ne sont pas requises sur les données à charger dans la base de données.

Ce composant peut être utilisé en stand alone ou en tant que composant de sortie.
### Paramètres

Le tableau suivant liste les paramètres utilisables dans la table **Advanced options**, dans l’onglet **Advanced settings**.

<table>
<thead>
<tr>
<th>Paramètre</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-If</td>
<td>Nom du fichier de log à générer. Les logs seront ajoutés à la suite du fichier de log, s’il existe déjà. Si le paramètre n’est pas spécifié, le nom par défaut du fichier de log est ‘&lt;table_name&gt;.&lt;db_name&gt;.nzlog’. Il est généré dans le répertoire de travail courant dans lequel le Job est exécuté.</td>
</tr>
<tr>
<td>-bf</td>
<td>Nom du fichier de mauvais enregistrements à générer. Le fichier de mauvais enregistrements contient tous les enregistrements ne pouvant être chargés, à cause d’une erreur interne Netezza. Les enregistrements seront ajoutés à la suite si le fichier de mauvais enregistrements existe déjà. Si le paramètre n’est pas spécifié, le nom par défaut du fichier de mauvais enregistrements est ‘&lt;table_name&gt;.&lt;db_name&gt;.nzbad’. Il est généré dans le répertoire de travail courant, dans lequel le Job est exécuté.</td>
</tr>
<tr>
<td>-outputDir</td>
<td>Chemin d’accès au répertoire dans lequel les fichiers de log et de mauvais enregistrements sont générés. Si le paramètre n’est pas spécifié, les fichiers seront générés sous le répertoire courant dans lequel le Job est exécuté.</td>
</tr>
<tr>
<td>-logFileSize</td>
<td>Taille maximale du fichier de log. La valeur est en MB. La valeur par défaut est de 2000 ou 2GB. Pour économiser de l’espace disque, spécifiez un petite valeur si votre Job est souvent exécuté.</td>
</tr>
<tr>
<td>-compress</td>
<td>Sélectionnez cette option si le fichier de données est compressée. Les valeurs valides sont “TRUE” et “FALSE”. La valeur par défaut est “FALSE”. Cette option n’est valide que si ce composant est utilisé par lui-même et non connecté à un autre composant via un flux d’entrée.</td>
</tr>
<tr>
<td>-skipRows &lt;n&gt;</td>
<td>Nombre de lignes à ignorer au début du fichier de données. Paramétrez la valeur à “1” si vous souhaitez ignorer la ligne d’en-tête du fichier d’entrée. La valeur par défaut est “0”. Cette option ne doit être sélectionnée que si le composant est utilisé par lui-même et non connecté à un autre composant via un flux d’entrée.</td>
</tr>
<tr>
<td>-maxRows &lt;n&gt;</td>
<td>Nombre de lignes à charger à partir du fichier d’entrée. Cette option ne doit être sélectionnée que si le composant est utilisé par lui-même et non connecté à un autre composant via un flux d’entrée.</td>
</tr>
<tr>
<td>-maxErrors</td>
<td>Nombre maximal d’enregistrements d’erreurs à autoriser avant de terminer le processus de chargement. La valeur par défaut est “1”.</td>
</tr>
</tbody>
</table>
### -ignoreZero

Les zéros contenus dans une source binaire généreront des erreurs. Paramétrez cette option à "NO" pour générer des erreurs ou à "YES" pour ignorer les bits ayant la valeur zéro. La valeur par défaut est "NO".

### -requireQuotes

Cette option nécessite que toutes les valeurs soient entourées de guillemets. La valeur par défaut est "FALSE". Cette option ne fonctionne pas avec un flux d'entrée pour le moment. Utilisez cette option uniquement en mode standalone avec un fichier existant.

### -nullValue <token>

Spécifiez le jeton pour indiquer une valeur null dans le fichier de données. La valeur par défaut est "NULL". Afin d'améliorer légèrement les performances, vous pouvez configurer cette valeur comme un champ vide en spécifiant la valeur : "".

### -fillRecord

Traite les derniers champs d'entrée manquants comme des null. Vous n'avez pas besoin de spécifier de valeur pour cette option dans le champ du tableau. Cette option n'est pas activée par défaut, les champs d'entrée devront donc correspondre exactement à toutes les colonnes de la table par défaut.

Les derniers champs d'entrée doivent être nullable dans la base de données.

### -ctrlChar

Accepte les caractères de contrôle dans les champs char/varchar (vous devrez échapper les caractères NUL, CR et LF). Vous n'avez pas besoin de spécifier de valeur pour cette option dans le champ de valeur du tableau. Cette option est désactivée par défaut.

### -ctInString

Accepte le caractère CR non échappé dans des champs char/varchar (LF devient une fin de ligne). Vous n'avez pas besoin de spécifier de valeur pour cette option dans le champ de valeur du tableau. Cette option est désactivée par défaut.

### -truncString

Tronque toute valeur de chaîne de caractères qui dépasse sa valeur déclarée de stockage char/varchar. Vous n'avez pas besoin de spécifier de valeur pour cette option dans le champ de valeur du tableau. Cette option est désactivée par défaut.

### -dateStyle

Spécifiez le format de la date des données d'entrée. Les valeurs valides sont : "YMD", "Y2MD", "DMY", "DMY2", "MDY", "MDY2", "MONDY", "MONDY2". La valeur par défaut est "YMD".

Le format de date dans la colonne du schéma du composant doit correspondre à la valeur spécifiée ici. Par exemple, si vous souhaitez charger une colonne DATE, spécifiez ainsi le format de date dans le schéma du composant : "yyyy-MM-dd" et paramétrez l'option -dateStyle à "YMD".

Pour une description plus précise du chargement de champs de dates et de temps, consultez [Charger des dates](#).
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dateDelim</td>
<td>Caractère séparateur entre les parties d’une date. La valeur par défaut est “-” pour tous les styles de date, sauf pour ‘MONDY[2]” pour lequel elle est “ ” (espace vide). <strong>Remarque :</strong> Le format de date dans la colonne du schéma du composant doit correspondre à la valeur spécifiée ici.</td>
</tr>
<tr>
<td>-y2Base</td>
<td>Première année exprimable via l’utilisation du style de date à deux chiffres (Y2).</td>
</tr>
</tbody>
</table>
| -timeStyle   | Spécifiez le format du temps dans les données d'entrée. Les valeurs valides sont : ‘24HOUR’ et ‘12HOUR’. La valeur par défaut est ‘24HOUR’. Pour améliorer légèrement les performances, gardez la valeur par défaut. Le format de temps dans la colonne du schéma du composant doit correspondre à la valeur spécifiée ici. Par exemple, si vous souhaitez charger une colonne de type TIME, configurez le format de date dans le schéma du composant à ’HH:mm:ss” et l’option -timeStyle à ’24HOUR”.
Pour une description plus précise du chargement de champs de dates et de temps, consultez *Charger des colonnes de type DATE, TIME et TIMESTAMP* à la page 2852. |
| -timeDelim   | Caractère séparateur des parties d’une donnée de temps. La valeur par défaut est “:”. Le format de temps dans la colonne du schéma du composant doit correspondre à la valeur spécifiée ici.                                                                                                                                                      |
| -timeRoundNanos | Autorise mais arrondit les nombres différents de zéro plus petits qu'une microseconde.                                                                                                                                                                                                                                                    |
| -boolStyle   | Spécifiez le format dans lequel les données de type Boolean sont écrites dans les données. Les valeurs valides sont : ”1_0”, ”T_F”, ”Y_N”, ”TRUE_FALSE”, ”YES”. La valeur par défaut est ”1_0”. Pour améliorer légèrement les performances, gardez la valeur par défaut.                                                                                   |
| -allowRelay  | Permet de continuer le chargement malgré une ou plusieurs réinitialisation(s) ou échec(s) des SPU (Snippet Processing Unit). Par défaut, ce comportement n’est pas autorisé.                                                                                                                                                                            |
| -allowRelay <n> | Spécifiez le nombre de continuations du chargement autorisé. La valeur par défaut est ”1”.                                                                                                                                                                                                                                          |
Charger des colonnes de type DATE, TIME et TIMESTAMP

Lorsque ce composant est utilisé avec un flux d’entrée, le format de date spécifié dans le schéma du composant doit correspondre à la valeur spécifiée pour les options -dateStyle, -dateDelim, -timeStyle, et -timeDelim.

<table>
<thead>
<tr>
<th>Type de données</th>
<th>Format de date dans le schéma</th>
<th>-dateStyle</th>
<th>-dateDelim</th>
<th>-timeStyle</th>
<th>-timeDelim</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>&quot;yyyy-MM-dd&quot;</td>
<td>&quot;YMD&quot;</td>
<td>&quot;.&quot;</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>TIME</td>
<td>&quot;HH:mm:ss&quot;</td>
<td>n/a</td>
<td>n/a</td>
<td>&quot;24HOUR&quot;</td>
<td>&quot;:&quot;</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>&quot;yyyy-MM-dd HH:mm:ss&quot;</td>
<td>&quot;YMD&quot;</td>
<td>&quot;.&quot;</td>
<td>&quot;24HOUR&quot;</td>
<td>&quot;:&quot;</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du composant **tMysqlOutputBulkExec**.
tNetezzaOutput

Ce composant écrit, met à jour, modifie ou supprime des entrées dans une base de données Netezza.
Le tNetezzaOutput exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Propriétés du tNetezzaOutput Standard

Ces propriétés sont utilisées pour configurer le tNetezzaOutput s’exécutant dans le framework de Jobs Standard.
Le composant tNetezzaOutput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>![Icon]</td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.</td>
</tr>
<tr>
<td>Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>
| ![Icon] | Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par |
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :
- Default : n’effectuer aucune opération de table.
- Drop and create the table : supprimer la table puis en créer une nouvelle.
- Create a table : créer une table qui n’existe pas encore.
- Create table if doesn’t exist : créer la table si nécessaire.
- Drop a table if exists and create : supprimer la table si elle existe déjà, puis en créer une nouvelle.
- Clear a table : supprimer le contenu de la table. |
| Action on data | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :
- Insert : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
- Update : Mettre à jour les entrées existantes.
- Insert or update : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. |
**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

**Delete** : Supprimer les entrées correspondant au flux d’entrée.

**Avertissement** :
Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

---

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
Die on error

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects.

Advanced settings

<table>
<thead>
<tr>
<th>Additional JDBC parameters</th>
<th>Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case Use an existing connection est décochée dans les Basic settings.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque :</strong> Vous pouvez appuyer sur Ctrl+Espace afin d'accéder à une liste de variables globales prédéfinies.</td>
</tr>
<tr>
<td>Use Batch</td>
<td>Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong> Cette case est disponible uniquement si vous avez sélectionné l’option Insert, Update ou Delete dans la liste Action on data.</td>
</tr>
<tr>
<td>Batch Size</td>
<td>Spécifiez le nombre d’enregistrements à traiter dans chaque lot.</td>
</tr>
<tr>
<td></td>
<td>Ce champ est disponible uniquement lorsque la case Use batch mode est cochée.</td>
</tr>
<tr>
<td>Commit every</td>
<td>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</td>
</tr>
<tr>
<td>Additional Columns</td>
<td>Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</td>
</tr>
<tr>
<td></td>
<td><strong>Name</strong> : Saisissez le nom de la colonne à modifier ou à insérer.</td>
</tr>
</tbody>
</table>
**SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.

**Position** : Sélectionnez Before, Replace ou After, en fonction de l’action à effectuer sur la colonne de référence.

**Reference column** : Saisissez une colonne de référence que le composant tNetezzaOutput peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

**Use field options** : Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.

**tStatCatcher Statistics** : Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong> : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong> : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong> : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong> : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_REJECTED</strong> : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des</td>
</tr>
</tbody>
</table>
netezzaOutput

requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données Netezza. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Limitation**


**Scénarios associés**

Pour un scénario associé au composant tNetezzaOutput, consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667 du composant tMysqlOutput.
**tNetezzaRollback**

Ce composant annule le commit de transaction dans la base de données Netezza connectée, afin d’éviter le commit de transaction involontaire.

**Propriétés du tNetezzaRollback Standard**

Ces propriétés sont utilisées pour configurer le tNetezzaRollback s’exécutant dans le framework de Jobs Standard.

Le composant tNetezzaRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tNetezzaConnection dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec d’autres composants Netezza, notamment les composants tNetezzaConnection et tNetezzaCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez...</td>
</tr>
</tbody>
</table>
pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénario associé

Pour un scénario associé au composant tNetezzaRollback, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623.
tNetezzaRow

Ce composant exécute des requêtes SQL déclarées sur la base de données Netezza spécifiée.

Selon la nature de la requête et de la base de données, tNetezzaRow agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Le tNetezzaRow est le composant spécifique à ce type de base de données. Il Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tNetezzaRow Standard

Ces propriétés sont utilisées pour configurer le tNetezzaRow s'exécutant dans le framework de Jobs Standard.

Le composant tNetezzaRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
</tbody>
</table>
Table Name | Nom de la table à traiter.
---|---
Query type | La requête peut être **Built-in** ou distante dans le **Repository**

**Built-in** : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.

**Repository** : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.

Query | Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.

Die on error | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

| Additional JDBC parameters | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.

| Propagate QUERY's recordset | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.

**Remarque :**

Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type **Object**. Ce composant est généralement suivi du tParseRecordSet.

| Use PreparedStatement | Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ **Query** de l'onglet **Basic settings**.

**Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL.

**Parameter Type** : Saisissez le type du paramètre.

**Parameter Value** : Saisissez la valeur du paramètre.

**Remarque :**
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

**Commit every**

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

- **QUERY** : requête traitée. Cette variable est une variable `Flow` et retourne une chaîne de caractères.
- **NB_EFFECTED** : nombre de lignes établies. Cette variable est une variable `After` et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case “Die on error” est décochée, si le composant a cette option.

Une variable `Flow` fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.

**Dynamic settings**

Cliquez sur le bouton `+` pour ajouter une ligne à la table. Dans le champ `Code`, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case “Use an existing connection” est cochée.
Dans la vue **Basic settings**, lorsqu'un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

**Limitation**

Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l'onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l'onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d'informations concernant l'installation des modules externes dans Talend Help Center (https://help.talend.com).

### Scénario associé

Pour un scénario associé au composant **tNetezzaRow**, consultez :

- **Scénario : Combiner deux flux pour une sortie sélective** à la page 2706.
- **Scénario : Supprimer et re-générer un index de table MySQL** à la page 2700.
**tNetezzaSCD**

Ce composant reflète et traque les modifications d’une table SCD Netezza dédiée.

Le tNetezzaSCD répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.

### Propriétés du tNetezzaSCD Standard

Ces propriétés sont utilisées pour configurer le tNetezzaSCD s’exécutant dans le framework de Jobs Standard.

Le composant tNetezzaSCD Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td>1.</td>
<td>Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
</tr>
</tbody>
</table>
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée. Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée : None : n'effectuer aucune opération de table. Create a table : créer une nouvelle table. Create table if not exists : si nécessaire, créer une table qui n'existe pas encore.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles : • View schema : sélectionnez cette option afin de voir le schéma. • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales. • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
</tbody>
</table>
**Built-in**: Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

**Repository**: Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

### SCD Editor

L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension.

Pour plus d’informations, consultez Méthodologie de gestion du SCD à la page 2716.

### Use memory saving Mode

Cochez cette case afin d’optimiser les performances du système.

### Source keys include Null

Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null.

**Avertissement :**

Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.

### Die on error

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez.

### Advanced settings

**Additional JDBC parameters**

Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case *Use an existing connection* est décochée dans les Basic settings.

**Remarque :**

Vous pouvez appuyer sur Ctrl+Espace afin d’accéder à une liste de variables globales prédéfinies.

**End date time details**

Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est 12:00:00.

Ce champ apparaît uniquement lorsqu’un SCD de Type 2 est utilisé et lorsque *Fixed year value* est sélectionné pour créer la date de fin du SCD.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Debug mode**

Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.
## Variables globales

| Global Variables | NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier. |
|                 | NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier. |
|                 | NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier. |
|                 | ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |

Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

## Utilisation

| Règle d'utilisation | Ce composant est utilisé comme composant de sortie. Il nécessite un composant d'entrée et un composant de sortie. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend. |

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu'un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de
Scénario associé

Pour un scénario associé, consultez tMysqlSCD à la page 2712.
tNetsuiteConnection

Ce composant crée une connexion au serveur SOAP NetSuite, pour pouvoir la réutiliser dans les autres composants NetSuite du Job.

**tNetsuiteConnection Standard properties**

Ces propriétés sont utilisées pour configurer le tNetsuiteConnection s’exécutant dans le framework de Standard.

Le composant tNetsuiteConnection Standard appartient à la famille Business.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint</td>
<td>Saisissez l’URL du Service Web requise pour vous connecter au serveur de NetSuite.</td>
</tr>
<tr>
<td>API version</td>
<td>Sélectionnez une version de l’API correspondant à la version de NetSuite utilisée.</td>
</tr>
<tr>
<td>E-mail et Password</td>
<td>Saisissez vos informations d’authentification pour accéder au service SOAP NetSuite.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Role</td>
<td>Saisissez l’ID de votre rôle NetSuite.</td>
</tr>
<tr>
<td>Account</td>
<td>Saisissez votre numéro de compte NetSuite pour accéder aux services Web (également appelé ID du compte).</td>
</tr>
<tr>
<td>Application ID</td>
<td>Saisissez l’ID de l’application NetSuite générée pour votre compte.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable customizations</td>
<td>Cochez cette case pour demander au composant de récupérer les personnalisations, à savoir les types d’enregistrements personnalisés et les champs personnalisés. Si cette case est cochée, les types d’enregistrements personnalisés sont récupérés et peuvent être sélectionnés dans la boîte de dialogue Record Type. Les champs personnalisés sont quant à eux récupérés et inclus dans le schéma.</td>
</tr>
</tbody>
</table>
Par défaut, cette case est cochée. Décochez-la si vous n'avez pas besoin des types d'enregistrements personnisés et des champs personnalisés.
Si cette case est cochée dans le composant tNetsuiteConnection, elle le sera également pour tous les autres composants utilisant cette connexion.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé avec les composants NetSuite.

### Scénario associé

Aucun scénario n’est disponible pour ce composant.
Ce composant invoque le service SOAP NetSuite et récupère les données selon les conditions spécifiées.

**Propriétés du tNetsuiteInput Standard**

Ces propriétés sont utilisées pour configurer le tNetsuiteInput s’exécutant dans le framework de Jobs Standard.

Le composant tNetsuiteInput Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th><strong>Connection Component</strong></th>
<th>Sélectionnez un composant tNetsuiteConnection pour réutiliser ses informations de connexion afin de configurer la connexion au serveur NetSuite. Lorsqu’un tNetsuiteConnection est sélectionné, les paramètres relatifs à la configuration de la connexion ne sont pas affichés.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Endpoint</strong></td>
<td>Saisissez l’URL du Service Web requise pour vous connecter au serveur de NetSuite.</td>
</tr>
<tr>
<td><strong>API version</strong></td>
<td>Sélectionnez une version de l’API correspondant à la version de NetSuite utilisée.</td>
</tr>
<tr>
<td><strong>E-mail et Password</strong></td>
<td>Saisissez vos informations d’authentification pour accéder au service SOAP NetSuite. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Role</strong></td>
<td>Saisissez l’ID de votre rôle NetSuite.</td>
</tr>
<tr>
<td><strong>Account</strong></td>
<td>Saisissez votre numéro de compte NetSuite pour accéder aux services Web (également appelé ID du compte).</td>
</tr>
<tr>
<td><strong>Record Type</strong></td>
<td>Pour ouvrir une boîte de dialogue, cliquez sur le bouton [...] à côté du champ. Sélectionnez un type d’enregistrement à récupérer ou cochez la case <strong>Use</strong></td>
</tr>
</tbody>
</table>
**custom object** et spécifiez un type d’enregistrement personnalisé dans le champ **Object Name**.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Avertissement** :

Le schéma de ce composant est renseigné de manière dynamique avec des colonnes correspondant aux champs des services NetSuite, qui varient selon le type d’enregistrement (**Record Type**) sélectionné. Vous pouvez supprimer les colonnes ne concernant pas votre opération, mais ne renommez pas de colonne du schéma, cela pourrait causer des erreurs d’exécution.

### Search conditions

Définissez les conditions de recherche afin de faire correspondre les champs pour récupérer les données. Cliquez sur le bouton `[+]` pour ajouter autant de lignes que nécessaire, chaque ligne pour une condition et définir chacune de vos conditions :

- **Field** : sélectionnez un champ dans lequel rechercher selon la condition spécifiée.
- **Operator** : sélectionnez un opérateur pour la condition de recherche.
- **Value** : saisissez la valeur pour la condition de recherche.
• **Value 2** : saisissez la seconde valeur de la condition de la recherche, si approprié, selon l’opérateur sélectionné.

### Advanced settings

| Body fields only | Cochez cette case pour récupérer uniquement les champs de type body.  
Décrochez cette case afin de récupérer les champs de type body et les valeurs des sous-listes. |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Enable customizations | Cochez cette case pour demander au composant de récupérer les personnalisations, à savoir les types d’enregistrements personnalités et les champs personnalités. Si cette case est cochée, les types d’enregistrements personnalités sont récupérés et peuvent être sélectionnés dans la boîte de dialogue **Record Type**. Les champs personnalités sont quant à eux récupérés et inclus dans le schéma.  
Par défaut, cette case est cochée. Décrochez-la si vous n’avez pas besoin des types d’enregistrements personnalisés et des champs personnalités. |
| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Variables globales

| Variables globales | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable *After* et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  
Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début dans un flux Talend et nécessite un composant de sortie. |
Scénario : Gestion des données avec NetSuite

Ce scénario décrit un Job lisant des données de NetSuite, transforme ces données et écrit les données transformées dans NetSuite, puis affiche les données transformées dans la console.

Ajouter les composants

Créez un nouveau Job et ajoutez les cinq composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette :

- deux **tNetsuitelInput** pour lire les données répondant aux critères de recherche spécifiés dans le serveur NetSuite.
- un **tMap** pour transformer les données d’entrée et leur donner la structure de sortie.
- un **tNetsuiteOutput** pour écrire les données transformées dans le serveur NetSuite.
- un **tLogRow** pour afficher les données reçues du serveur NetSuite.

Configurer et relier les composants

Lire, transformer et écrire des données dans NetSuite

Procédure

1. Double-cliquez sur le premier **tNetsuitelInput** pour ouvrir sa vue **Basic settings**.
2. Dans les champs Endpoint, API version, E-mail, Password, Role, Account, et Application ID, saisissez les informations requises pour accéder à NetSuite.

3. Cliquez sur le bouton [...] à côté du champ Record Type. Dans la boîte de dialogue qui s'ouvre, sélectionnez Location afin de lire depuis NetSuite les informations relatives au type d'emplACEMENT.

Le schéma correspondant au type d'emplACEMENT de l'enregistrement est automatiquement récupéré.

4. Définissez une condition de recherche.
   a) Cliquez sur le bouton [+] sous la table Search conditions pour ajouter une ligne.
   b) Cliquez sur la cellule dans la colonne Field et sélectionnez internalId dans la liste déroulante.
   c) Cliquez dans la cellule de la colonne Operator et sélectionnez List - Any Of dans la liste déroulante.
   d) Dans la cellule de la colonne Value, saisissez java.util.Arrays.asList("1") pour récupérer uniquement l'enregistrement dont l'internalId est 1.

5. Connectez le premier tNetsuiteInput au tMap à l'aide d'un lien Row > Main.

6. Double-cliquez sur le tNetsuiteOutput pour ouvrir sa vue Basic settings.


9. Cliquez sur le bouton [...] à côté du champ **Record Type**. Dans la boîte de dialogue qui s’ouvre, sélectionnez **Location** afin d’écrire depuis NetSuite les informations relatives au type d’emplacement.

10. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur de schéma. Vous pouvez voir que le schéma est pré-renseigné. Laissez uniquement les colonnes correspondant à votre opération et supprimez les autres. Dans cet exemple, conservez les colonnes **Name**, **Parent**, **IncludeChildren**, **IsInactive**, **TranPrefix**, **Logo**, **MakeInventoryAvailable**, **MakeInventoryAvailableStore**, **CustomFieldList** et **ExternalId**.
11. Connectez le tMap au tNetsuiteOutput à l'aide d'un lien Row > "New Output" (Main). Dans la boîte de dialogue qui s'ouvre, saisissez le nom de la connexion de sortie, out dans cet exemple. Cliquez sur Yes afin que le tMap récupère le schéma du composant cible tNetsuiteOutput.

12. Double-cliquez sur le tMap pour ouvrir sa vue Map Editor et configurer les mappings de flux de données.

   ![Map Editor](image)

   a) Dans la table de sortie out, configurez la valeur de la colonne Name en saisissant "Paris" dans le champ correspondant de la colonne Expression.

   b) Glissez-déposez la colonne Parent dans la table d'entrée row1 dans la colonne Expression de la ligne correspondante de la table out.

   c) Glissez-déposez les autres colonnes IncludeChildren, IsInactive, TranPrefix, Logo, MakeInventoryAvailable, MakeInventoryAvailableStore, CustomFieldList et ExternalId de la table d'entrée row1 dans la colonne Expression de la ligne correspondante de la table de sortie out.

Récupérer les données transformées de NetSuite

**Procédure**

1. Double-cliquez sur le second tNetsuiteInput pour ouvrir sa vue Basic settings.

3. Cliquez sur le bouton [...] à côté du champ **Record Type**. Dans la boîte de dialogue qui s'ouvre, sélectionnez **Location** afin de lire depuis NetSuite les informations relatives au type d'emplACEMENT.

Le schéma correspondant au type d'emplACEMENT de l'enregistrement est automatiquement récupéré.

4. Définissez une condition de recherche.
   a) Cliquez sur le bouton [+ ] sous la table **Search Criteria** pour ajouter une ligne.
   b) Cliquez dans la cellule de la colonne **Field** et sélectionnez **name** dans la liste déroulante.
   c) Cliquez dans la cellule de la colonne **Operator** et sélectionnez **String - Contains** dans la liste déroulante.
   d) Dans la cellule de la colonne **Value**, saisissez **Paris** pour récupérer uniquement l'enregistrement dont le nom (**name**) contient **Paris**.

5. Double-click **tLogRow** to open its **Basic settings** view.
6. Dans la zone **Mode**, sélectionnez **Vertical (each row is a key/value list)** pour une lisibilité optimale des résultats.

7. Connectez le second tNetsuitInput au tLogRow à l’aide d’un lien **Row > Main**.

8. Connectez le premier tNetsuitInput au second tNetsuitInput à l’aide d’un lien **Trigger > OnSubjobOk**.

**Sauvegardez et exécutez le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder le Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour exécuter le Job.

Comme montré dans la capturer d’écran ci-dessus les données ont été transformées et écrites dans NetSuite, puis les données transformées ont été récupérées de NetSuite et affichées dans la console.
tNetsuiteOutput

Ce composant invoque le service SOAP Netsuite et insère, met à jour ou supprime des données sur le serveur SOAP NetSuite.

Propriétés du tNetsuiteOutput Standard

Ces propriétés sont utilisées pour configurer le tNetsuiteOutput s’exécutant dans le framework de Jobs Standard.
Le composant tNetsuiteOutput Standard appartient aux familles Business et Cloud.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Connection Component</th>
<th>Sélectionnez un composant tNetsuiteConnection pour réutiliser ses informations de connexion afin de configurer la connexion au serveur NetSuite. Lorsqu’un tNetsuiteConnection est sélectionné, les paramètres relatifs à la configuration de la connexion ne sont pas affichés.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint</td>
<td>Saisissez l’URL du Service Web requise pour vous connecter au serveur de NetSuite.</td>
</tr>
<tr>
<td>API version</td>
<td>Sélectionnez une version de l’API correspondant à la version de NetSuite utilisée.</td>
</tr>
<tr>
<td>E-mail et Password</td>
<td>Saisissez vos informations d’authentification pour accéder au service SOAP NetSuite. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Role</td>
<td>Saisissez l’ID de votre rôle NetSuite.</td>
</tr>
<tr>
<td>Account</td>
<td>Saisissez votre numéro de compte NetSuite pour accéder aux services Web (également appelé ID du compte).</td>
</tr>
<tr>
<td>Record Type</td>
<td>Pour ouvrir une boîte de dialogue, cliquez sur le bouton [...] à côté du champ. Sélectionnez un type d’enregistrement à récupérer ou cochez la case Use</td>
</tr>
</tbody>
</table>
custom object et spécifiez un type d’enregistrement personnalisé dans le champ Object Name.

### Schema et Edit schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

**Avertissement** :
Le schéma de ce composant est pré-renseigné avec des colonnes correspondant aux champs des services NetSuite, qui varient selon le type d’enregistrement (Record Type) sélectionné. Vous pouvez supprimer les colonnes ne concernant pas votre opération, mais ne renommez pas de colonne du schéma, cela pourrait causer des erreurs d’exécution.

### Action
 Sélectionnez une action à effectuer dans la liste.
• **Add** : ajouter un nouvel enregistrement à vos données. Si un doublon est trouvé et que la case **Die on error** est cochée, le Job s’arrête. Si la case **Die on error** est décochée, le doublon est ignoré et le Job continue.

• **Update** : apporter des modifications à un enregistrement existant.

Si l’enregistrement correspondant à l’ID interne (Internal ID) spécifié n’existe pas et que la case **Die on error** est cochée, le Job s’arrête.

Pour utiliser cette option, la colonne **InternalId** doit exister dans le schéma et être définie comme Key.

• **Upsert** : mettre à jour l’enregistrement correspondant à l’ID spécifié ou insérer un nouvel enregistrement s’il n’existe pas.

Pour utiliser cette option, la colonne pour l’ID de l’enregistrement doit exister dans le schéma et être définie comme clé (Key).

• **Delete** : supprimer un enregistrement existant.

Si l’enregistrement correspondant à l’ID interne (Internal ID) spécifié n’existe pas et que la case **Die on error** est cochée, le Job s’arrête.

Pour utiliser cette option, la colonne **InternalId** doit exister dans le schéma et être définie comme Key.

### Sync outgoing schema

Après la mise à jour du schéma du composant, cliquez sur ce bouton pour propager le schéma au flux de sortie, qui peut être un flux normal, un flux de rejet, ou bien les deux, si le composant n’en a aucun.

### Advanced settings

#### Enable customizations

Cochez cette case pour demander au composant de récupérer les personnalisations, à savoir les types d’enregistrements personnalisés et les champs personnalisés. Si cette case est cochée, les types d’enregistrements personnalisés sont récupérés et peuvent être sélectionnés dans la boîte de dialogue **Record Type**. Les champs personnalisés sont quant à eux récupérés et inclus dans le schéma.

Par défaut, cette case est cochée. Décocochez-la si vous n’avez pas besoin des types d’enregistrements personnalisés et des champs personnalisés.

#### Use native Upsert operation

Cochez cette case pour utiliser l’opération Upsert fournie par NetSuite.

Pour utiliser l’opération Upsert de NetSuite, la colonne **ExternalId** doit exister dans le schéma et être définie comme clé (Key).

Cette option est disponible lorsque l’action **Upsert** est sélectionnée dans la liste **Action** dans l’onglet **Basic settings**.
Pour des raisons de compatibilité, cette case est décochée par défaut. Le composant effectue l'action Add ou Update pour une action Upsert selon si l'ID de l’enregistrement est spécifié.

<table>
<thead>
<tr>
<th>Batch size</th>
<th>Spécifiez le nombre d’enregistrements à traiter dans chaque lot. Notez que l’option de traitement par lots est désactivée, si le composant de sortie dispose d’un flux de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CURRENT_INTERNALID : ID interne de l’enregistrement en cours d’insertion ou de traitement. Cette variable est une variable Flow et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de fin dans un flux Talend et nécessite un composant d’entrée. |

**Scénario associé**

Pour un scénario associé, consultez Scénario : Gestion des données avec NetSuite à la page 2876.
tNormalize

Ce composant normalise un flux entrant en fonction du standard SQL pour améliorer la qualité des données et faciliter leur mise à jour.

Propriétés du tNormalize Standard

Ces propriétés sont utilisées pour configurer le tNormalize s'exécutant dans le framework de Jobs Standard.
Le composant tNormalize Standard appartient à la famille Processing.
Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
| | • View schema : sélectionnez cette option afin de voir le schéma.
| | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
| | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |
| Column to normalize | Sélectionnez la colonne du flux entrant sur laquelle est basée la normalisation. |
| Item separator | Renseignez le séparateur délimitant les données du flux entrant. |

Remarque :
Le séparateur d’éléments se base sur des expressions régulières. Par conséquent, le caractère "." (caractère spécial dans les expressions régulières) doit être évité ou utilisé avec prudence.

### Advanced settings

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get rid of duplicated rows from output</td>
<td>Cochez cette case pour supprimer les lignes doublons du flux de sortie.</td>
</tr>
<tr>
<td>Use CSV parameters</td>
<td>Cochez cette case pour prendre en compte les paramètres spécifiques aux fichiers CSV, notamment la manière de protéger les caractères dans le champ escape mode et le type de guillemet dans le champ enclosure.</td>
</tr>
<tr>
<td>Discard the trailing empty strings</td>
<td>Cochez cette case pour ignorer les chaînes de caractères vides à la fin d’un groupe d’éléments.</td>
</tr>
<tr>
<td>Trim resulting values</td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ des données de résultat.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsque les cases Discard the trailing empty string et Trim resulting values sont cochées, le traitement de la première option s’effectue en premier.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant intermédiaire dans un flux de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

**Scénario : Normaliser des données**

Ce scénario illustre un Job qui normalise une liste de mots-clés des sujets d’un forum puis affiche les résultats dans un tableau dans la console de la vue **Run**.

Cette liste n’est pas vraiment organisée et contient des espaces vides en début en fin de champs, ainsi que des mots-clés répétés.

ldap, db2, jdbc driver, grid computing, talend architecture, content, environment,, tmap,, eclipse, database, java,postgresql, tmap, database, java, sybase, deployment,, repository, database, informix, java

**Construire le Job**

**Procédure**

1. Cliquez et déposez les composants suivants dans l’espace de modélisation : **tFileInputDelimited**, **tNormalize**, **tLogRow**.

2. Reliez les composants à l’aide de liens **Row > Main**.

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le composant **tFileInputDelimited**, pour ouvrir sa vue **Basic settings**.
2. Dans le champ **File name**, spécifiez le chemin d’accès au fichier d’entrée à normaliser.

3. Cliquez sur le bouton [...] à côté du champ **Edit schema** afin d’ouvrir la boîte de dialogue [Schema] et configurez le schéma d’entrée en ajoutant une colonne nommée **Tags**. Cela fait, cliquez sur **OK** pour valider votre schéma et fermer la boîte de dialogue. Laissez les autres paramètres tels qu’ils sont.

4. Double-cliquez sur le composant **tNormalize** pour ouvrir sa vue **Basic settings**.

5. Vérifiez le schéma et, si nécessaire, cliquez sur le bouton **Sync columns** afin de synchroniser le schéma et le schéma du composant d’entrée.

6. Définissez la colonne sur laquelle est basée la normalisation.
Dans ce scénario, le schéma d'entrée possède une seule colonne, *Tags*. Acceptez donc les paramètres par défaut.

7. Dans la vue **Advanced settings**, cochez les cases **Get rid of duplicate rows from output**, **Discard the trailing empty strings** et **Trim resulting values**.

8. Dans les propriétés du composant **tLogRow**, sélectionnez le mode **Table (print values in the cells of table)**.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.
2. Cliquez sur le bouton **Run**, dans la vue **Run** ou appuyez sur **F6** pour exécuter votre Job.

**Résultats**

La liste est nettoyée, les doublons, et les espaces vides sont supprimés. Le résultat est affiché dans un tableau dans la console.
tOleDbInput

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

Le tOleDbInput lit une base de données et en extrait des champs à l’aide de requêtes. Il offre la flexibilité de la requête en base de données et couvre toutes les possibilités de requête SQL.

Propriétés du tOleDbInput Standard

Ces propriétés sont utilisées pour configurer le tOleDbInput s’exécutant dans le framework de Jobs Standard.

Le composant tOleDbInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schema and Edit schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td><strong>Built-In</strong></td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Edit schema</strong></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td><strong>View schema</strong></td>
<td>sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td><strong>Change to built-in property</strong></td>
<td>sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td><strong>Update repository connection</strong></td>
<td>sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la</td>
</tr>
<tr>
<td>Table Name</td>
<td>Saisissez le nom de table à traiter.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Query Type</td>
<td>Sélectionnez le type de requête <strong>Built-in</strong> ou <strong>Repository</strong> dans la liste déroulante.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez la requête appropriée dans le <strong>Repository</strong>. Le champ <strong>Query</strong> est renseigné automatiquement.</td>
</tr>
<tr>
<td>Guess Query</td>
<td>Cliquez sur le bouton <strong>Guess Query</strong> pour générer la requête correspondant au schéma de la table définie dans le champ <strong>Query</strong>.</td>
</tr>
<tr>
<td>Guess schema</td>
<td>Cliquez sur le bouton <strong>Guess schema</strong> pour récupérer le schéma de la table.</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Spécifiez la requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| Trim all the String/Char columns | Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères. |
| Trim column | Cochez cette case dans la colonne **Trim** pour supprimer les espaces en début et en fin de champ dans les colonnes correspondantes. Ce champ apparaît uniquement lorsque la case **Trim all the String/Char columns** est décochée. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| **Global Variables** | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier. **QUERY** : requête traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères. **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. |
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espacement** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le [Guide utilisateur du Studio Talend](#).

### Utilisation

| Règle d’utilisation                  | Ce composant est utilisé en tant que composant d’entrée d’un Job et nécessite un lien de sortie. |

### Scénario associé

Pour un scénario similaire, consultez le [tMysqlInput](#) à la page 2631.
tOleDbOutput

Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Ce composant écrit, modifie ou supprime les données d’une base de données.

Propriétés du tOleDbOutput Standard

Ces propriétés sont utilisées pour configurer le tOleDbOutput s’exécutant dans le framework de Jobs Standard.

Le composant tOleDbOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
</tr>
</thead>
</table>

| Table |
| Spécifiez le nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois. |

| Action on data |
| Dans la liste déroulante, sélectionnez parmi les opérations suivantes celle que vous souhaitez effectuer sur les données. |
| • Insert : insérer des données. |
| • Update : mettre à jour des données. |
| • Insert or update : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. |
| • Update or insert : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré. |
| • Delete : supprimer des données. |

**Avertissement :**

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations Update et Delete. Pour cela, cliquez sur le bouton […] à côté du champ Edit Schema et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet Advanced settings pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case Use field options et cochez la case Key in update correspondant à la colonne...
<table>
<thead>
<tr>
<th>Clear data in table</th>
<th>Cochez cette case pour supprimer les données dans la table définie avant toute opération.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema and Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonne) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td>Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
<tr>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
<td></td>
</tr>
<tr>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
<td></td>
</tr>
<tr>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
<td></td>
</tr>
<tr>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content.</td>
<td></td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>Additional columns</th>
<th>Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name : Saisissez le nom de la colonne à modifier ou à insérer.</td>
<td></td>
</tr>
<tr>
<td>SQL expression : Saisissez la déclaration SQL à exécuter pour modifier ou insérer des données dans les colonnes correspondantes.</td>
<td></td>
</tr>
<tr>
<td>Position : Sélectionnez Before, After ou Replace, en fonction de l’action à effectuer sur la colonne de référence.</td>
<td></td>
</tr>
</tbody>
</table>
• **Reference column** : Saisissez une colonne de référence utilisée pour situer ou remplacer une nouvelle colonne ou celle à modifier.

Cette option n'est pas disponible si vous venez de créer la table de base de données (même si vous la supprimez avant).

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use field options</strong></td>
<td>Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.</td>
</tr>
<tr>
<td><strong>Field options</strong></td>
<td>Cochez cette ou ces case(s) dans la table si vous souhaitez utiliser la ou les colonne(s) de schéma correspondantes en tant que base pour votre opération.</td>
</tr>
</tbody>
</table>

  - **Key in update** : définir la clé primaire à utiliser lors d'une opération de mise à jour.
  - **Key in delete** : définir la clé primaire à utiliser lors d'une opération de suppression.
  - **Updatable** : définir la ou les colonne(s) à mettre à jour.
  - **Insertable** : définir la ou les colonne(s) à insérer.

Ce champ apparaît uniquement lorsque la case **Use field options** est cochée.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable debug mode</strong></td>
<td>Cochez cette case pour afficher chaque étape du processus d'écriture dans la base de données.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong> : nombre de lignes traitées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong> : nombre de lignes mises à jour. Cette variable est une variable <em>After</em> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong> : nombre de lignes insérées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong> : nombre de lignes supprimées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_REJECTED</strong> : nombre de lignes rejetées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>QUERY</strong> : requête traitée. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères.</td>
<td></td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong> : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
<td></td>
</tr>
</tbody>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant est normalement utilisé en tant que composant de sortie d’un Job et nécessite un lien d’entrée. |

**Scénario associé**

Pour un scénario similaire, consultez le tMysqlOutput à la page 2661.
tOleDbRow

Ce composant agit sur la structure même de la base de données ou sur les données, selon la nature de la requête et de la base de données.

Le tOleDbRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée.

Propriétés du tOleDbRow Standard

Ces propriétés sont utilisées pour configurer le tOleDbRow s’exécutant dans le framework de Jobs Standard.

Le composant tOleDbRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td>Built-In</td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Repository</td>
<td>Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
<td></td>
</tr>
<tr>
<td>• View schema</td>
<td>sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td>• Change to built-in property</td>
<td>sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td>• Update repository connection</td>
<td>sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content.</td>
</tr>
<tr>
<td>Table Name</td>
<td>Nom de la table à traiter.</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Query type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez la requête appropriée dans le Repository. Le champ <em>Query</em> est renseigné automatiquement.</td>
</tr>
<tr>
<td>Guess Query</td>
<td>Cliquez sur le bouton <em>Guess Query</em> pour générer la requête correspondant au schéma de votre table dans le champ <em>Query</em>.</td>
</tr>
<tr>
<td>Query</td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Rejects.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| Propagate QUERY’s recordset | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste *use column*. |
|                           | **Remarque** : |
|                           | Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type *Object*. Ce composant est généralement suivi du tParseRecordSet. |

|                       | **Parameter Index** : Saisissez la position du paramètre dans l’instruction SQL. |
|                       | **Parameter Type** : Saisissez le type du paramètre. |
|                       | **Parameter Value** : Saisissez la valeur du paramètre. |
|                       | **Remarque** : |
|                       | Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance. |
**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.</td>
</tr>
<tr>
<td></td>
<td>Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les</td>
</tr>
</tbody>
</table>
paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

**Scénario associé**

Pour un scénario similaire, consultez *tMysqlRow* à la page 2696.
tOpenbravoERPInput

Ce composant extrait des données d’une base OpenBravoERP selon certaines conditions définies dans des colonnes spécifiques.

Le tOpenbravoERPInput se connecte à une entité de la base de données OpenbravoERP via le service Web adéquat.

Propriétés du tOpenbravoERPInput Standard

Ces propriétés sont utilisées pour configurer le tOpenbravoERPInput s’exécutant dans le framework de Jobs Standard.

Le composant tOpenbravoERPInput Standard appartient à la famille Business.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Openbravo REST WebService URL</th>
<th>Saisissez l’URL du service Web permettant de se connecter à la base de données OpenbravoERP.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d’authentification de l’utilisateur au service Web.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Entity</td>
<td>Sélectionnez l’entité adéquate à partir de la liste déroulante.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la</td>
</tr>
</tbody>
</table>
métadonnée du schéma dans la fenêtre [Repository Content].
Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent.

Remarque :
Pour ce composant, le schéma correspond à l'entité sélectionnée.

<table>
<thead>
<tr>
<th>WHERE Clause</th>
<th>Saisissez votre clause WHERE.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order by</td>
<td>Cochez cette case afin de sélectionner comment ordonner les résultats (les éléments de la liste déroulante dépendent de l'entité sélectionnée).</td>
</tr>
<tr>
<td></td>
<td><strong>Sort</strong> : Dans cette liste, vous pouvez choisir de trier vos résultats de manière ascendante (Ascending) ou descendante (Descending).</td>
</tr>
<tr>
<td>First result</td>
<td>Saisissez le numéro de la ligne que vous souhaitez récupérer en premier.</td>
</tr>
<tr>
<td>Max result</td>
<td>Saisissez le nombre maximal de résultats que vous souhaitez récupérer.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Advanced separator (for numbers)</th>
<th>Cochez cette option pour modifier les séparateurs utilisés pour les nombres :</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Thousands separator</strong> : définissez le séparateur utilisé pour les milliers.</td>
</tr>
<tr>
<td></td>
<td><strong>Decimal separator</strong> : définissez le séparateur utilisé pour les décimaux.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire.</th>
</tr>
</thead>
</table>

**Scénario associé**

Pour un scénario associé, consultez *Scénario : Écrire des données dans une base de données Microsoft CRM et attribuer des conditions aux colonnes pour extraire des lignes spécifiques* à la page 2383.
**tOpenbravoERPOutput**

Ce composant écrit des données dans une base de données OpenbravoERP.

Le `tOpenbravoERPOutput` écrit dans un objet de la base de données OpenbravoERP via le service Web adéquat.

**Propriétés du `tOpenbravoERPOutput Standard`**

Ces propriétés sont utilisées pour configurer le `tOpenbravoERPOutput` s'exécutant dans le framework de Jobs Standard.

Le composant `tOpenbravoERPOutput Standard` appartient à la famille Business.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th><strong>Openbravo REST Webservice URL</strong></th>
<th>Saisissez l'URL du service Web permettant de se connecter à la base de données OpenbravoERP.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez les informations d'authentification de l'utilisateur au service Web.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <code>Password</code>, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Action on data</strong></td>
<td>Sélectionnez dans la liste si vous souhaitez mettre à jour ou créer des données (Update/Create), ou si vous préférez en supprimer (Remove).</td>
</tr>
<tr>
<td><strong>Use existing data file</strong></td>
<td>Cochez cette case afin de sélectionner votre fichier en parcourant votre répertoire.</td>
</tr>
<tr>
<td><strong>Entity</strong></td>
<td>Sélectionnez l'entité adéquate à partir de la liste déroulante.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
</tbody>
</table>
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>NB_LINE</strong> : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <strong>After</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>NB_LINE_UPDATED</strong> : nombre de lignes mises à jour. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_LINE_CREATED</strong> : nombre de lignes créées. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_LINE_REMOVED</strong> : nombre de lignes supprimées. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_LINE_UNAUTHORIZED</strong> : nombre de lignes non autorisées. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_LINE_FAILED</strong> : nombre de lignes en erreur. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.
Utilisation

| Règle d'utilisation | Ce composant est utilisé comme composant de sortie. Il nécessite un composant d’entrée. |

Scénario associé

Pour un scénario associé, consultez Scénario : Écrire des données dans une base de données Microsoft CRM et attribuer des conditions aux colonnes pour extraire des lignes spécifiques à la page 2383.
**tOracleBulkExec**

Ce composant améliore les performances au cours des opérations effectuées sur les données d’une base de données Oracle.

Le composant tOracleBulkExec insère, ajoute, remplace ou supprime les données d’une base de données Oracle.

Les composants tOracleOutputBulk et tOracleBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tOracleOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

**Propriétés du tOracleBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tOracleBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tOracleBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par...
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

**Connection type**

Liste déroulante des pilotes disponibles:

**Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l’interface d’appel de la base de données Oracle (Oracle Call Interface) accompagnée d’un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle.

**Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l’alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante.

**Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.

**Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.

**DB Version**

 Sélectionnez la version d’Oracle que vous utilisez.

**Host**

Adresse IP du serveur de base de données.

**Port**

 Numéro du port d’écoute du serveur de base de données.

**Database**

Nom de la base de données.

**Schema**

Nom du schéma.

**Username et Password**

 Informations d’authentification de l’utilisateur de base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Table**

Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois.

**Action on table**

Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :

**None** : n’effectuer aucune opération de table.
<table>
<thead>
<tr>
<th>Data file name</th>
<th>Nom du fichier à charger.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Avertissement</strong> :</td>
</tr>
<tr>
<td></td>
<td>Ce fichier doit être situé sur la même machine que le serveur de base de données.</td>
</tr>
<tr>
<td>Action on data</td>
<td>Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td><strong>Insert</strong> : Insérer des données dans une table vide. Si la table contient déjà des données, le Job s'arrête et les données existantes restent dans la table.</td>
</tr>
<tr>
<td></td>
<td><strong>Update</strong> : Mettre à jour les données existantes. Il est nécessaire de configurer la clé du schéma lorsque cette option est sélectionnée.</td>
</tr>
<tr>
<td></td>
<td><strong>Append</strong> : Ajoute des données à la table, que celle-ci soit vide ou non.</td>
</tr>
<tr>
<td></td>
<td><strong>Replace</strong> : Si la table contient déjà des données, supprime toutes les données existantes et insère les nouvelles données. Si la table est vide, insère les nouvelles données.</td>
</tr>
<tr>
<td></td>
<td><strong>Truncate</strong> : Si la table contient déjà des données, tronque toutes les données existantes et insère les nouvelles données. Si la table est vide, insère les nouvelles données.</td>
</tr>
<tr>
<td></td>
<td><strong>Avertissement</strong> : Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations d’Update. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une utilisation de base, ou à partir des options avancées (Advanced settings) pour une utilisation optimisée de ces opérations.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
</tbody>
</table>
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Advanced settings

| Advanced separator (for numbers) | Cochez cette option pour modifier les séparateurs utilisés pour les nombres :
|                                | **Thousands separator** : définissez le séparateur utilisé pour les milliers.
|                                | **Decimal separator** : définissez le séparateur utilisé pour les décimaux. |

| Use existing control file | Cochez cette case si vous utilisez un fichier de contrôle (.ctl) et spécifiez son chemin d’accès dans le champ **.ctl file name**. |

| Record format | Définissez le format de l’enregistrement :
|              | **Default** : les paramètres du format sont définis par défaut.
|              | **Stream** : déterminez le séparateur de fin d’enregistrement.
|              | **Fixed** : déterminez la longueur d’enregistrement.
|              | **Variable** : déterminez la taille du champ spécifiant la longueur de l’enregistrement. |

<p>| Specify .ctl file’s INTO TABLE clause manually | Cochez cette case pour renseigner manuellement la clause INTO TABLE du fichier de contrôle, directement dans le code. |</p>
<table>
<thead>
<tr>
<th><strong>Fields terminated by</strong></th>
<th>Caractère, chaîne ou expression régulière séparant les champs :</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>aucun caractère de séparation.</td>
</tr>
<tr>
<td>Whitespace</td>
<td>le caractère de séparation est un espace.</td>
</tr>
<tr>
<td>EOF (used for loading LOBs from lobfile)</td>
<td>la séparation est constituée d’un caractère de fin de fichier (End Of File).</td>
</tr>
<tr>
<td>Other terminator</td>
<td>la séparation est constituée d’un caractère autre que ceux précédemment énoncés.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Use fields enclosure</strong></th>
<th>Cochez cette case si vous voulez utiliser des caractères pour entourer le texte :</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields enclosure (left part)</td>
<td>caractère délimitant les champs sur la gauche.</td>
</tr>
<tr>
<td>Field enclosure (right part)</td>
<td>caractère délimitant les champs sur la droite.</td>
</tr>
</tbody>
</table>

| **Use schema's Date Pattern to load Date field** | Cochez cette case pour utiliser le modèle de date contenu dans le schéma pour remplir le champ concernant la date. |

| **Specify field condition** | Cochez cette case pour définir une condition de chargement des données. |

| **Preserve blanks** | Cochez cette case pour conserver les espaces vides. |

| **Trailing null columns** | Cochez cette case pour charger des données avec des colonnes vides. |

<table>
<thead>
<tr>
<th><strong>Load options</strong></th>
<th>Cliquez sur le bouton [+] afin d’ajouter des options de chargement des données :</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>choisissez l’un des paramètres de chargement à partir de la liste déroulante.</td>
</tr>
<tr>
<td>Value</td>
<td>définissez la valeur à attribuer au paramètre.</td>
</tr>
</tbody>
</table>

| **NLS Language** | A partir de la liste déroulante, définissez la langue utilisée pour les données qui ne sont pas codées en Unicode. |

| **Set Parameter NLS_TERRITORY** | Cochez cette case pour modifier les conventions d’appellation utilisées pour la numérotation des jours et des semaines. La valeur par défaut est celle du système d’exploitation. |

| **Encoding** | Sélectionnez l’encodage dans la liste ou saisissez-le entre guillemets doubles si le type d’encodage utilisé n’existe pas dans la liste. Ce champ est obligatoire pour la gestion de données de bases de données. |

<table>
<thead>
<tr>
<th><strong>Output</strong></th>
<th>Sélectionnez la sortie à utiliser pour récupérer la sortie standard de la base Oracle :</th>
</tr>
</thead>
<tbody>
<tr>
<td>to console</td>
<td>vers la console.</td>
</tr>
<tr>
<td>to global variable</td>
<td>vers la variable globale.</td>
</tr>
<tr>
<td>Convert columns and table names to uppercase</td>
<td>Cochez cette case pour mettre le nom des colonnes et celui de la table en majuscules.</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE_DATA : nombre de lignes lues. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_BAD : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>RETURN_CODE : code retour indiquant le résultat du traitement. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant dédié offre performance et flexibilité lors de requêtes sur les bases de données Oracle.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre</td>
</tr>
</tbody>
</table>
Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 264 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

### Limitation
Le serveur/client de la base de données doit être installé sur la même machine que le Studio Talend ou que le Job contenant un tOracleBulkExec afin que le composant fonctionne correctement.

### Scénario : Supprimer et insérer des données dans une base Oracle

Ce scénario décrit comment supprimer le contenu d'une base de données Oracle et insérer de nouvelles données à partir du contenu d'un fichier d'entrée. Le Job est constitué de trois composants qui créent le contenu, le transfèrent dans un fichier qui sera ensuite chargé dans la base Oracle après suppression des données de la table.

#### Construire le Job

**Procédure**

1. Cliquez et déposez les composants suivants de la Palette dans l'espace de modélisation graphique : tOracleInput, tFileOutputDelimited et tOracleBulkExec.
2. Connectez le composant tOracleInput au tFileOutputDelimited à l'aide d'un lien Row > Main.
3. Reliez le tOracleInput au tOracleBulkExec à l'aide d'un lien OnSubjobOk.

#### Résultats
Configurer les composants

Procédure

1. Paramétrez les informations de connexion à la base de données Oracle. Il est conseillé de stocker ces informations dans une métadonnée du Repository pour pouvoir les réutiliser par la suite dans tous vos Jobs.

2. Paramétrez le schéma, s'il n'est pas conservé dans le Repository. Dans cet exemple, le schéma contient les quatre colonnes suivantes : ID_Contract, ID_Client, Contract_type et Contract_Value.

3. Dans la liste Encoding, sélectionnez l’option Custom et définissez AL32UTF8 comme encodage.

4. Dans la vue Basic settings du composant tFileOutputDelimited, paramétrez ses propriétés, notamment les champs File Name, Row separator et Field separator avec respectivement le chemin d’accès et le nom du fichier, le séparateur de lignes et de champs.

5. Double-cliquez ensuite sur le composant tOracleBulkExec pour définir les propriétés d’alimentation de la base de données.
6. Dans la liste **Property Type**, sélectionnez le mode **Repository** si vous avez stocké les informations de connexion à la base de données sous le nœud **Metadata** du **Repository** ou sélectionnez le mode **Built-In** pour les définir manuellement. Dans ce scénario, utilisez le mode **Built-In**.

7. Configurez si nécessaire les paramètres de connexion dans les champs **Host**, **Port**, **Database**, **Schema**, **Username** et **Password**.

8. Dans le champ **Table**, renseignez le nom de la table à alimenter et dans le champ **Action on data**, sélectionnez l'action à effectuer, ici, **Insert**.

9. Dans la liste **Schema**, sélectionnez le mode **Built-In** puis cliquez sur le bouton `[...]` à côté du champ **Edit Schema** pour définir les données à passer au composant suivant.


11. Cochez la case **Use an existing control file** si vous avez stocké les informations relatives à l’état de la base de données dans un fichier de contrôle (.ctl). Sinon, renseignez, si nécessaire, les champs suivants manuellement : **Record format**, **Specify .ctl file's INTO TABLE clause manually**, **Field terminated by**, **Use field enclosure**, **Use schema's Date Pattern to load Date field**, **Specify field condition**, **Preserve blanks**, **Trailing null columns**, **Load options**, **NLS Language** et **Set Parameter NLS_TERRITORY** en fonction de votre base de données.

12. Dans la liste **Encoding**, sélectionnez le type d’encodage ou saisissez-le entre guillemets doubles s’il n’est pas dans la liste.


**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** pour exécuter le Job. Le log s’affiche dans la console de la vue **Run** et la table est alimentée par les données du fichier paramétré.

**Résultats**

Voir également : **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692.
tOracleClose

Ce composant ferme une connexion à la base de données Oracle connectée.

**Propriétés du tOracleClose Standard**

Ces propriétés sont utilisées pour configurer le tOracleClose s'exécutant dans le framework de Jobs Standard.

Le composant tOracleClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tOracleConnection dans la liste.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Oracle, notamment tOracleConnection et tOracleCommit. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. |
Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tOracleCommit

Ce composant valide les données traitées dans un Job à partir d’une base de données connectée.

Le tOracleCommit utilise une connexion unique pour commiter en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

**Propriétés du tOracleCommit Standard**

Ces propriétés sont utilisées pour configurer le tOracleCommit s’exécutant dans le framework de Jobs Standard.

Le composant tOracleCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tOracleConnection dans la liste.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. <strong>Avertissement :</strong> Si vous utilisez un lien de type Row &gt;Main pour relier le tOracleCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants Oracle et notamment <strong>tOracleConnection</strong> et <strong>tOracleRollback</strong>.</th>
</tr>
</thead>
</table>
| **Dynamic settings** | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  
  Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.  
  Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez  Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**. |

### Scénario associé

Pour un scénario associé au composant **tOracleCommit**, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tOracleConnection

Ce composant ouvre une connexion à la base de données Oracle spécifiée afin de pouvoir la réutiliser dans le ou les sous-job(s) suivant(s).

Propriétés du tOracleConnection Standard

Ces propriétés sont utilisées pour configurer le tOracleConnection s’exécutant dans le framework de Jobs Standard.

Le composant tOracleConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Connection type</td>
<td>Liste déroulante des pilotes disponibles :</td>
</tr>
<tr>
<td></td>
<td>Oracle OCI : Sélectionnez ce type de connexion pour utiliser l’interface d’appel de la base de données Oracle (Oracle Call Interface) accompagnée d’un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle.</td>
</tr>
<tr>
<td></td>
<td>Oracle Service Name : Sélectionnez ce type de connexion pour utiliser l’alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante.</td>
</tr>
<tr>
<td></td>
<td>Oracle SID : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.</td>
</tr>
<tr>
<td></td>
<td>Oracle Custom : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version d’Oracle que vous utilisez.</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Use tns file</td>
<td>Cochez cette case pour utiliser les métadonnées d’un contexte contenu dans un fichier Tns.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Un même fichier Tns peut contenir plusieurs contextes.</td>
</tr>
<tr>
<td></td>
<td><strong>TNS File :</strong> Renseignez manuellement le chemin d’accès ou cliquez sur le bouton [...] pour parcourir vos dossiers jusqu’au fichier Tns à utiliser.</td>
</tr>
<tr>
<td></td>
<td><strong>Select a DB Connection in Tns File :</strong> Cliquez sur le bouton [...] pour afficher tous les contextes du fichier Tns et sélectionner celui à utiliser.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Use SSL</td>
<td>Cochez cette case pour utiliser le protocole SSL pour la protection des communications avec le serveur de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété est disponible uniquement lorsque ORACLE CUSTOM est sélectionnée dans la liste déroulante <strong>Connection Type</strong> et que Oracle 12-7 est sélectionnée dans la liste déroulante <strong>DB Version</strong>.</td>
</tr>
<tr>
<td>TrustStore file</td>
<td>Chemin vers le fichier TrustStore servant à authentifier votre serveur de base de données lors de l’authentification SSL.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété est disponible uniquement si la case <strong>Use SSL</strong> est cochée.</td>
</tr>
<tr>
<td>TrustStore password</td>
<td>Mot de passe du fichier TrustStore.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété est disponible uniquement si la case <strong>Use SSL</strong> est cochée.</td>
</tr>
<tr>
<td>Propriété</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Need Client authentication</strong></td>
<td>Cochez cette case uniquement lorsqu’une authentification client SSL est requise. Cette propriété est disponible uniquement si la case <strong>Use SSL</strong> est cochée.</td>
</tr>
<tr>
<td><strong>KeyStore file</strong></td>
<td>Chemin vers le fichier KeyStore servant à l’authentification client SSL. Cette propriété n’est disponible que lorsque la case <strong>Need Client authentication</strong> est cochée.</td>
</tr>
<tr>
<td><strong>KeyStore password</strong></td>
<td>Mot de passe du fichier KeyStore. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres. Cette propriété n’est disponible que lorsque la case <strong>Need Client authentication</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Disable CBC protection</strong></td>
<td>Cochez cette case pour désactiver l’utilisation de CBC (Cipher Block Chaining, enchaînement des blocs de chiffrement) dans l’authentification SSL. Cette propriété est disponible uniquement si la case <strong>Use SSL</strong> est cochée.</td>
</tr>
</tbody>
</table>
| **Additional JDBC parameters** | Spécifiez des informations supplémentaires de connexion à la base de données créée.  
**Remarque :**  
Dans ce champ, vous pouvez paramétrer le type d’encodage. |
| **Use or register a shared DB Connection** | Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, petits ou parents.  
Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un composant **tRunJob** ayant une de ces options activée fera échouer votre Job.  
Cette case est indisponible lorsque la case **Specify a data source alias** est cochée. |
| **Specify a data source alias** | Cochez cette case et spécifiez l’alias de la source de données créée dans **Talend Runtime** pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans **Talend Runtime**. Pour voir un exemple de cas d’utilisation, |
Advanced settings

**Auto Commit**

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne committe pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job au niveau du Job ainsi qu’au niveau de chaque composant.

Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé avec des composants Oracle, notamment les composants `tOracleCommit` et `tOracleRollback`.

Scénario associé

Pour un scénario associé au composant `tOracleConnection`, consultez `tMysqlConnection` à la page 2618.
tOracleInput

Ce composant lit une base de données et en extrait des champs à l’aide de requêtes.
Le tOracleInput exécute une requête de base de données dans laquelle l’ordre doit strictement correspondre à l’ordre défini dans le schéma. Puis la liste des champs est transmise au composant via une connexion Main Row.

Propriétés du tOracleInput Standard

Ces propriétés sont utilisées pour configurer le tOracleInput s’exécutant dans le framework de Jobs Standard.
Le composant tOracleInput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
</table>
| Property type | Peut être Built-In ou Repository.  
Built-In : propriétés utilisées ponctuellement.  
Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant. |
|  | Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.  
Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend. |
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
Remarque :  
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par |
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th><strong>Connection type</strong></th>
<th>Liste déroulante des pilotes disponibles:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oracle OCI</strong></td>
<td>Sélectionnez ce type de connexion pour utiliser l’interface d’appel de la base de données Oracle (Oracle Call Interface) accompagnée d’un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle.</td>
</tr>
<tr>
<td><strong>Oracle Custom</strong></td>
<td>Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.</td>
</tr>
<tr>
<td><strong>Oracle Service Name</strong></td>
<td>Sélectionnez ce type de connexion pour utiliser l’alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante.</td>
</tr>
<tr>
<td><strong>WALLET</strong></td>
<td>Sélectionnez ce type de connexion pour stocker les informations d’authentification dans un portefeuille (wallet) Oracle.</td>
</tr>
<tr>
<td><strong>Oracle SID</strong></td>
<td>Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.</td>
</tr>
</tbody>
</table>

| **DB Version** | Sélectionnez la version d’Oracle que vous utilisez. |
| **Host**      | Adresse IP du serveur de base de données. |
| **Port**      | Numéro du port d’écoute du serveur de base de données. |
| **Database**  | Nom de la base de données. |
| **Oracle schema** | Nom du schéma Oracle. |

<table>
<thead>
<tr>
<th><strong>Username et Password</strong></th>
<th>Informations d’authentification de l’utilisateur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, |
évitez le mot réservé **line** lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th>Table name</th>
<th>Nom de la table de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query type</strong> et <strong>Query</strong></td>
<td>Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.</td>
</tr>
<tr>
<td><strong>Specify a data source alias</strong></td>
<td>Cochez cette case et spécifiez l’alias de la source de données créée dans <strong>Talend Runtime</strong> pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans <strong>Talend Runtime</strong>. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans <strong>Talend Runtime</strong> pour récupérer les données d’une base de données <strong>MySQL</strong> à la page 2647. Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l’alias de la source de données spécifié. Cette option est indisponible lorsque la case <strong>Use an existing connection</strong> est cochée.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| **Additional JDBC parameters** | Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. |
Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, 
encryption=1;clientname=Talend.

Ce champ n’est pas disponible si la case Use an existing connection est cochée.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Use cursor
Cochez cette case et, dans le champ Cursor size qui s’affiche, spécifiez le nombre de lignes à extraire en une fois dans la base de données. Les performances peuvent être optimisées en configurant la taille d’extraction à une valeur appropriée.

### Trim all the String/Char columns
Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.

### Trim column
Supprime les espaces en début et en fin de champ dans les colonnes sélectionnées.

### No null values
Cochez cette case pour optimiser les performances s’il n’y a pas de valeur nulle.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Oracle.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de</td>
</tr>
</tbody>
</table>
contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Limitation**


**Scénario : Utiliser les paramètres de contexte lors de la lecture d’une table d’une base de données Oracle**

Dans ce scénario, une table est lue dans une base de données Oracle, à l’aide d’un paramètre de contexte pour référer au nom de la table.

**Déposer et relier les composants**

**Procédure**

1. Créez un nouveau Job et ajoutez les composants suivants en saisissant leur nom dans l’espace de modélisation graphique et en les déposant depuis la Palette : un **tOracleInput** et un **tLogRow**.
2. Reliez le **tOracleInput** au **tLogRow** via un lien Row > Main.
Configurer les composants

Procédure

1. Double-cliquez sur le tOracleInput pour ouvrir sa vue Basic settings.

![Image of tOracleInput configuration settings]

2. Sélectionnez une connexion dans la liste déroulante Connection Type. Dans cet exemple, saisissez OracleSID.

Sélectionnez la version de la base de données Oracle à utiliser dans la liste déroulante DB Version. Dans cet exemple, saisissez Oracle 12.7.

Dans le champ Host, saisissez l’adresse IP du serveur de la base de données Oracle, "192.168.31.32".

Dans le champ Database, saisissez le nom de la base de données, "TALEND" dans cet exemple.

Dans le champ Oracle schema, saisissez le nom du schéma Oracle, "TALEND" dans cet exemple.

Dans les champs Username et Password, saisissez les informations d’authentification.

4. Cliquez sur le bouton pour ajouter quatre colonnes : ID et AGE de type Integer et NAME et SEX, de type String.
   Cliquez sur OK pour fermer l'éditeur du schéma et acceptez la propagation proposée par la boîte de dialogue qui s'ouvre.

5. Placez votre souris sur le champ Table Name et appuyez sur F5 pour configurer les paramètres de contexte. La boîte de dialogue [New Context Parameter] s'ouvre.

Pour plus d'informations concernant les paramètres de contexte, consultez le Guide utilisateur du Studio Talend.

6. Dans le champ Name, laissez le nom du paramètre de contexte, TABLE dans cet exemple.
Dans le champ Default value, saisissez le nom de la table de base de données Oracle à interroger. Dans cet exemple, saisissez PERSON.

7. Cliquez sur Finish pour valider.

La paramètre de contexte context.TABLE apparaît automatiquement dans le champ Table Name.

8. Dans la liste Query Type, sélectionnez Built-In. Cliquez sur Guess Query pour obtenir l’instruction de la requête.

```
"SELECT "+context.TABLE+"."ID", "+context.TABLE+"."NAME", "+context.TABLE+"."SEX, "+context.TABLE+"."AGE FROM "+context.TABLE
```


10. Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour un meilleur affichage des résultats.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl+S afin de sauvegarder le Job.

2. Appuyez sur F6 pour exécuter le Job.

```
[statistics] connecting to socket on port 3750
[statistics] connected

  tLogRow_1

+----------+-------+-----+-----+
| ID | NAME  | SEX | AGE |
+----------+-------+-----+-----+
| 1   | Ford  | Male| 25  |
| 2   | Rose  | Female| 30 |
| 3   | Sabrina | Female| 28 |
| 4   | Teddy | Male| 32  |
| 5   | Kate  | Male| 35  |
+----------+-------+-----+-----+
[statistics] disconnected
```

Comme affiché dans la capture d’écran, les données dans la table de la base de données Oracle PERSON sont affichées dans la console.
Scénarios associés

Consultez les scénarios associés :

- Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520.
**tOracleOutput**

Ce composant écrit, met à jour, modifie ou supprime les données d’une base de données.

Le tOracleOutput exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

**Propriétés du tOracleOutput Standard**

Ces propriétés sont utilisées pour configurer le tOracleOutput s’exécutant dans le framework de Jobs Standard.

Le composant tOracleOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><img src="icon" alt="Configure connection" /></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Liste déroulante des pilotes disponibles:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle OCI</td>
<td>Oracle OCI : Sélectionnez ce type de connexion pour utiliser l’interface d’appel de la base de données Oracle (Oracle Call Interface) accompagnée d’un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle.</td>
</tr>
<tr>
<td>Oracle Custom</td>
<td>Oracle Custom : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.</td>
</tr>
<tr>
<td>Oracle Service Name</td>
<td>Oracle Service Name : Sélectionnez ce type de connexion pour utiliser l’alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante.</td>
</tr>
<tr>
<td>WALLET</td>
<td>WALLET : Sélectionnez ce type de connexion pour stocker les informations d’authentification dans un port efeuille (wallet) Oracle.</td>
</tr>
<tr>
<td>Oracle SID</td>
<td>Oracle SID : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.</td>
</tr>
</tbody>
</table>

| DB Version                       | Sélectionnez la version d’Oracle que vous utilisez. |
| Host                             | Adresse IP du serveur de base de données. |
| Port                             | Numéro du port d’écoute du serveur de base de données. |
| Database                         | Nom de la base de données. |
| Username et Password             | Informations d’authentification de l’utilisateur de la base de données. |
|                                  | Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Table                            | Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois. |
| Action on table                  | Remarque : |
La liste **Action on table** n’est pas disponible si vous cochez la case **Enable parallel execution** dans la vue **Advanced settings**.

Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :

- **Default** : n’effectuer aucune opération de table.
- **Drop and create table** : supprimer la table puis en créer une nouvelle.
- **Create table** : créer une table qui n’existe pas encore.
- **Create table if not exists** : créer la table si nécessaire.
- **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.
- **Clear table** : supprimer le contenu de la table.
- **Truncate table** : supprimer le contenu de la table, mais sans possibilité de Rollback.
- **Truncate table with reuse storage** : supprimer le contenu de la table. Il n’y a pas de possibilité de rollback. Cependant, vous pouvez réutiliser le stockage alloué à la table, même si le stockage est considéré comme vide.

**Avertissement** :

Si vous cochez la case **Use an existing connection** et que vous sélectionnez une option différente de **Default** dans la liste **Action on table**, une instruction de commit est générée automatiquement, avant l’opération d’insertion/mise à jour/suppression de données.

**Action on data**

Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

- **Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
- **Update** : Mettre à jour les entrées existantes.
- **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
- **Delete** : Supprimer les entrées correspondant au flux d’entrée.

**Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir...
simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Die on error

Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Specify a data source alias

Cochez cette case et spécifiez l'alias de la source de données créée dans **Talend Runtime** pour utiliser le pool.
de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour voir un exemple de cas d'utilisation, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d'une base de données MySQL à la page 2647.

Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l’alias de la source de données spécifié.

Cette option est indisponible lorsque la case Use an existing connection est cochée.

**Advanced settings**

| **Additional JDBC parameters** | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case Use an existing connection est décochée dans les Basic settings. |
| **Override any existing NLS_LANG environment variable** | Cochez cette case pour écraser les variables d’environnement NLS_LANG déjà définies. |
| **Commit every** | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |
| **Additional Columns** | Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier. |
| Name | Saisissez le nom de la colonne à modifier ou à insérer. |
| SQL expression | Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes. |
| Position | Sélectionnez Before, Replace ou After, en fonction de l’action à effectuer sur la colonne de référence. |
**Reference column :** Saisissez une colonne de référence que le composant **tOracleOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

**Use field options**
Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.

**Use Hint Options**
Cochez cette case pour ajouter des indicateurs (ou Hints) permettant d'optimiser le plan d'exécution de la requête afin d'obtenir les meilleurs performances possibles.
- **HINT** : spécifiez le hint dont vous avez besoin, à l'aide de la syntaxe /*+ */ .
- **POSITION** : spécifiez où positionner le hint dans une instruction SQL.
- **SQL STMT** : sélectionnez l'instruction SQL que vous devez utiliser.

**Convert columns and table to uppercase**
Cochez cette case pour passer les noms de colonnes et de tables en majuscule.

**Enable debug mode**
Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.

**Use Batch**
Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.

**Batch Size**
Spécifiez le nombre d’enregistrements à traiter dans chaque lot.
Ce champ est disponible uniquement lorsque la case **Use batch mode** est cochée.

**Support null in “SQL WHERE” statement**
Cochez cette case pour prendre en compte les valeurs Null lors d’une instruction WHERE.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>QUERY : requête traitée. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.

**Utilisation**

**Règle d'utilisation**

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL.

Ce composant doit être utilisé en tant que composant de sortie. Il permet de faire des actions sur une table ou les données d'une table d'une base de données Oracle. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur. Pour un exemple d'utilisation, consultez Scénario : Récupérer les données erronées à l'aide d’un lien Reject à la page 2675 du composant tMysqlOutput.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

**Limitation**

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour
Scénarios associés

Pour un scénario associé au composant **tOracleOutput**, consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le **tMysqlOutput** à la page 2667.
tOracleOutputBulk

Ce composant écrit un fichier composé de colonnes et basé sur le séparateur défini et sur les standards Oracle.

Les composants tOracleOutputBulk et tOracleBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tOracleOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Propriétés du tOracleOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tOracleOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tOracleOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom du fichier à générer.</td>
</tr>
<tr>
<td>Avertissement</td>
<td>Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.</td>
</tr>
<tr>
<td>Create directory if not exists</td>
<td>Cette option permet de créer le dossier contenant le fichier de sortie s’il n’existe pas déjà.</td>
</tr>
<tr>
<td>Append</td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
</tbody>
</table>
## Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Built-In

Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

### Repository

Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

## Advanced settings

### Advanced separator (for numbers)

Cochez cette option pour modifier les séparateurs utilisés pour les nombres :

- **Thousands separator** : définissez le séparateur utilisé pour les milliers.
- **Decimal separator** : définissez le séparateur utilisé pour les décimaux.

### Field separator

Caractère, chaîne ou expression régulière séparant les champs.

### Row separator

Chaîne (ex : `\n` sous Unix) séparant les lignes.

### Encoding Type

Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce
champ est obligatoire pour la manipulation des données de base de données.

<table>
<thead>
<tr>
<th>Bulk file parameters</th>
<th>Configurer les paramètres Buffer Size et StringBuilder Size afin d'optimiser les performances selon la mémoire.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

| Global Variables | NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.  

ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.  

Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.  

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec le composant tOracleBulkExec. Ensemble ils offrent un gain de performance important pour l'alimentation d'une base de données Oracle. |

### Scénarios associés

Pour des scénarios associés au composant tOracleOutputBulk, consultez les scénarios suivants :

- **Scénario :** Insérer des données transformées dans une base MySQL à la page 2685 du tMysqlOutputBulk.
- **Scénario :** Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec.
- **Scénario :** Supprimer et insérer des données dans une base Oracle à la page 2914 du tOracleBulkExec.
**tOracleOutputBulkExec**

Ce composant effectue une action d'Insert dans la base de données Oracle spécifiée.

En tant que composant dédié, il permet un gain de performance pendant les opérations d'Insert dans une base de données Oracle.

Les composants tOracleOutputBulk et tOracleBulkExec sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tOracleOutputBulkExec.

**Propriétés du tOracleOutputBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tOracleOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tOracleOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository..</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Tableau des connexions</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Connection type** | Liste déroulante des pilotes disponibles :
  - **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l’interface d’appel de la base de données Oracle (Oracle Call Interface) accompagnée d’un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle.
  - **Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l’alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante.
  - **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.
  - **Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters. |
| **DB Version** | Sélectionnez la version d’Oracle que vous utilisez. |
| **Host** | Adresse IP du serveur de la base de données.
  - Seuls `localhost`, `127.0.0.1` ou l’adresse IP exacte de la machine locale permettent un fonctionnement optimal.
  - Le serveur de la base de données doit être installé sur la même machine que le Studio Talend ou le Job contenant un `tOracleOutputBulkExec`. |
| **Port** | Numéro du port d’écoute du serveur de base de données. |
| **Database** | Nom de la base de données. |
| **Schema** | Nom du schéma. |
| **Username et Password** | Informations d’authentification de l’utilisateur de base de données.
  - Pour saisir le mot de passe, cliquez sur le bouton `[...]` à côté du champ `Password`, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur `OK` afin de sauvegarder les paramètres. |
| **Table** | Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit déjà exister pour que l’opération `d’Insert` soit autorisée. |
| **Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée : |
**None**: n'effectuer aucune opération de table.

**Drop and create table**: supprimer la table puis en créer une nouvelle.

**Create table**: créer une table qui n'existe pas encore.

**Create table if not exists**: créer la table si nécessaire.

**Drop table if exists and create**: supprimer la table si elle existe déjà, puis en créer une nouvelle.

**Clear table**: supprimer le contenu de la table.

**Truncate table**: supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Nom du fichier à générer et à charger.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Avertissement</strong>:</td>
<td>Ce fichier est généré sur la machine spécifiée par l'URI dans le champ Host et doit être sur la même machine que le serveur de la base de données.</td>
</tr>
</tbody>
</table>

| Create directory if not exists | Cette case est cochée par défaut. Cette option permet de créer un répertoire où stocker la table de sortie si elle n'existe pas. |

| Append | Cochez cette case afin d’ajouter de nouvelles lignes à la fin de l’enregistrement. |

<table>
<thead>
<tr>
<th>Action on data</th>
<th>Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Insert</strong>:</td>
<td>Insérer des données dans une table vide. Si la table contient déjà des données, le Job s’arrête et les données existantes restent dans la table.</td>
</tr>
<tr>
<td><strong>Update</strong>:</td>
<td>Mettre à jour les données existantes. Il est nécessaire de configurer la clé du schéma lorsque cette option est sélectionnée.</td>
</tr>
<tr>
<td><strong>Append</strong>:</td>
<td>ajoute des données à la table, que celle-ci soit vide ou non.</td>
</tr>
<tr>
<td><strong>Replace</strong>:</td>
<td>si la table contient déjà des données, supprime toutes les données existantes et insère les nouvelles données. Si la table est vide, insère les nouvelles données.</td>
</tr>
<tr>
<td><strong>Truncate</strong>:</td>
<td>Si la table contient déjà des données, tronque toutes les données existantes et insère les nouvelles données. Si la table est vide, insère les nouvelles données.</td>
</tr>
</tbody>
</table>

**Avertissement**:
Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations d’Update. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une utilisation de base, ou à partir des options avancées (Advanced settings) pour une utilisation optimisée de ces opérations.
### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Field separator

Caractère, chaîne de caractères ou expression régulière pour séparer les champs.

### Advanced settings

#### Advanced separator (for numbers)

Cochez cette option pour modifier les séparateurs utilisés pour les nombres :

- **Thousands separator** : définissez le séparateur utilisé pour les milliers.
- **Decimal separator** : définissez le séparateur utilisé pour les décimaux.

#### Use existing control file

Cochez cette case si vous utilisez un fichier de contrôle (.ctl).

#### Record format

Définissez le format de l’enregistrement :
| **Specify .ctl file’s INTO TABLE clause manually** | Cochez cette case pour renseigner manuellement la clause INTO TABLE du fichier de contrôle, directement dans le code. |
| **Fields terminated by** | Caractère, chaîne ou expression régulière séparant les champs :  
  - **None** : aucun caractère de séparation.  
  - **Whitespace** : le caractère de séparation est un espace.  
  - **EOF (used for loading LOBs from lobfile)** : la séparation est constituée d’un caractère de fin de fichier (End Of File).  
  - **Other terminator** : la séparation est constituée d’un caractère autre que ceux précédemment énoncés. |
| **Use fields enclosure** | Cochez cette case si vous voulez utiliser des caractères pour entourer le texte :  
  - **Fields enclosure (left part)** : caractère délimitant les champs sur la gauche.  
  - **Field enclosure (right part)** : caractère délimitant les champs sur la droite. |
| **Use schema’s Date Pattern to load Date field** | Cochez cette case pour utiliser le modèle de date contenu dans le schéma pour remplir le champ concernant la date. |
| **Specify field condition** | Cochez cette case pour définir une condition de chargement des données. |
| **Preserve blanks** | Cochez cette case pour conserver les espaces vides. |
| **Trailing null columns** | Cochez cette case pour charger des données avec des colonnes vides. |
| **Load options** | Cliquez sur le bouton [+] afin d’ajouter des options de chargement des données :  
  - **Parameter** : choisissez l’un des paramètres de chargement à partir de la liste déroulante.  
  - **Value** : définissez la valeur à attribuer au paramètre. |
| **NLS Language** | A partir de la liste déroulante, définissez la langue utilisée pour les données qui ne sont pas codées en Unicode. |
| **Set Parameter NLS_TERRITORY** | Cochez cette case pour modifier les conventions d’appellation utilisées pour la numérotation des jours et... |
des semaines. La valeur par défaut est celle du système d’exploitation.

**Encoding**

Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

**Oracle encoding type**

Sélectionnez le type d’encodage spécifique à Oracle pour le traitement des données. Ce type d’encodage permet au **tOracleOutputBulkExec** de générer correctement la commande Oracle de masse.

**Output**

Sélectionnez dans la liste déroulante l’endroit où rediriger le flux de sortie :

- **to console** : vers la console de log.
- **to global variable** : vers une variable globale.

**Convert columns and table names to uppercase**

Cochez cette case pour mettre le nom des colonnes et celui de la table en majuscules.

**Bulk file parameters**

Configurez les paramètres **Buffer Size** et **StringBuilder Size** afin d’optimiser les performances selon la mémoire.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Utilisation

**Règle d’utilisation**

Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans la base de données.

**Dynamic settings**

Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de...
paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

| Limitation | Le serveur/client de la base de données doit être installé sur la même machine que le Studio Talend ou que le Job contenant un tOracleOutputBulkExec, afin que le composant fonctionne correctement. |

Scénarios associés

Pour des scénarios associés au composant tOracleOutputBulkExec, consultez les scénarios suivants :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du tMysqlOutputBulk.
- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec.
- Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du tOracleBulkExec.
**tOracleRollback**

Ce composant annule le commit de transaction dans une base de données Oracle connectée pour éviter le commit involontaire de transaction.

**Propriétés du tOracleRollback Standard**

Ces propriétés sont utilisées pour configurer le tOracleRollback s'exécutant dans le framework de Jobs Standard.

Le composant tOracleRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tOracleConnection dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Oracle, notamment les composants tOracleConnection et tOracleCommit. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez... |
dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Pour un scénario associé au composant **tOracleRollback**, consultez **tMysqlRollback** à la page 2694.
tOracleRow

Ce composant exécute la requête SQL saisie sur la base de données Oracle spécifiée.

Selon la nature de la requête et de la base de données, le tOracleRow agit sur la structure même de la base de données ou sur les données (mais sans les manipuler). Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Le tOracleRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

**Propriétés du tOracleRow Standard**

Ces propriétés sont utilisées pour configurer le tOracleRow s’exécutant dans le framework de Jobs Standard.

Le composant tOracleRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

| Connection type | Liste déroulante des pilotes disponibles:  
Oracle OCI : Sélectionnez ce type de connexion pour utiliser l’interface d’appel de la base de données Oracle (Oracle Call Interface) accompagnée d’un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle.  
Oracle Service Name : Sélectionnez ce type de connexion pour utiliser l’alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante.  
Oracle SID : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.  
Oracle Custom : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version d’Oracle que vous utilisez.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma Oracle.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
</tbody>
</table>
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.  

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• View schema : sélectionnez cette option afin de voir le schéma. |
| **• Change to built-in property** | sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales. |
| **• Update repository connection** | sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content. |

| **Built-in** | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| **Repository** | Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |

| **Table name** | Nom de la table de base de données. |
| **Query type** | Peut être Built-in ou Repository. |

| **Built-in** | Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder. |
| **Repository** | Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement. |

| **Query** | Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma. |

| **Use NB_LINE_** | Cette option vous permet d'alimenter la variable afin de passer au sous-job ou au composant suivant le nombre de lignes mises à jour/supprimées/insérées. Ce champ s'applique uniquement si la requête saisie dans le champ Query est une requête INSERT, UPDATE ou DELETE. |

| **• NONE** | n'alimente pas la variable. |
| **• INSERTED** | alimente la variable avec le nombre de lignes insérées. |
| **• UPDATED** | alimente la variable avec le nombre de lignes mises à jour. |
| **• DELETED** | alimente la variable avec le nombre de lignes supprimées. |

| **Specify a data source alias** | Cochez cette case et spécifiez l'alias de la source de données créée dans Talend Runtime pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour voir un exemple de cas d'utilisation, consultez Scénario : Déploiement de votre Job dans... |
Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647.

Si vous utilisez la configuration de la base de données du composant, la connexion à votre source de données se ferme à la fin du composant. Pour empêcher la fermeture de la connexion, utilisez une connexion partagée à la base de données, avec l’alias de la source de données spécifié.

Cette option est indisponible lorsque la case **Use an existing connection** est cochée.

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

**Advanced settings**

**Additional JDBC parameters**

Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.

**Propagate QUERY’s recordset**

Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.

*Remarque :
Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type **Object**. Ce composant est généralement suivi du **tParseRecordSet**.*

**Use PreparedStatement**

Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des “?” dans l’instruction SQL définie dans le champ **Query** de l’onglet **Basic settings**.

- **Parameter Index** : Saisissez la position du paramètre dans l’instruction SQL.
- **Parameter Type** : Saisissez le type du paramètre.
- **Parameter Value** : Saisissez la valeur du paramètre.

*Remarque :
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance. Vous pouvez également utiliser les PreparedStatement afin d’éviter les injections SQL. Pour un scénario utilisant cette fonctionnalité, consultez Scénario 2 : Utiliser l’instance...*
### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez...</td>
</tr>
</tbody>
</table>
pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Pour un scénario associé, consultez :

- **Scénario : Combiner deux flux pour une sortie sélective** à la page 2706.
- **Procédure du tDBSQLRow**.
- **Scénario : Supprimer et re-générer un index de table MySQL** à la page 2700 du tMysqlRow.
- **Scénario 2 : Utiliser l’instance PreparedStatement pour faire une requête sur des données** à la page 2702 du tMysqlRow.
tOracleSCD

Ce composant reflète et traque les modifications d’une table SCD Oracle dédiée. Le tOracleSCD répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.

Propriétés du tOracleSCD Standard

Ces propriétés sont utilisées pour configurer le tOracleSCD s’exécutant dans le framework de Jobs Standard.

Le composant tOracleSCD Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

| Database | Sélectionnez un type de base de données dans la liste et cliquez sur Apply. |
| Property type | Peut être Built-in ou Repository. |
| Built-in | Propriétés utilisées ponctuellement. |
| Repository | Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

**Connection type**

Liste déroulante des pilotes disponibles:

- **Oracle OCI** : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (Oracle Call Interface) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle.

- **Oracle Service Name** : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante.

- **Oracle SID** : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.

- **Oracle Custom** : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.

**DB Version**

 Sélectionnez la version d'Oracle que vous utilisez.

**Host**

Adresse IP du serveur de base de données.

**Port**

 Numéro du port d'écoute du serveur de base de données.

**Database**

 Nom de la base de données.

**Schema**

Nom du schéma de la base de données.

**Username et Password**

Informations d'authentification de l'utilisateur de la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Table**

Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.

**Action on table**

Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :

- **None** : n'effectuer aucune opération de table.

- **Create a table** : créer une nouvelle table.

- **Create table if not exists** : si nécessaire, créer une table qui n'existe pas encore.

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés.
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


**SCD Editor**

L'éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension.

Pour plus d’informations, consultez Méthodologie de gestion du SCD à la page 2716.

**Use memory saving Mode**

Cochez cette case pour améliorer les performances du système.

**Source keys include Null**

Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs Null.

⚠️ **Avertissement** :

Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.

**Die on error**

Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et d’ignorer les lignes en erreur.

**Advanced settings**

**Additional JDBC parameters**

Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n’est pas disponible lorsque vous utilisez l’option **Use an existing connection** dans les **Basic settings**.
**End date time details**

Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format `HH:mm:ss`. La valeur par défaut pour ce champ est 12:00:00.

Ce champ apparaît uniquement lorsqu’un SCD de Type 2 est utilisé et lorsque **Fixed year value** est sélectionné pour créer la date de fin du SCD.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Debug mode**

Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.

---

### Variables globales

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>NB_LINE_UPDATED</strong> : nombre de lignes mises à jour. Cette variable est une variable <code>After</code> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>NB_LINE_INSERTED</strong> : nombre de lignes insérées. Cette variable est une variable <code>After</code> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td><strong>NB_LINE_REJECTED</strong> : nombre de lignes rejetées. Cette variable est une variable <code>After</code> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <code>After</code> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

---

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez</td>
</tr>
</tbody>
</table>
pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez **tMysqlSCD** à la page 2712.
tOracleSCDELT

Ce composant reflète et traque les modifications d’une table Oracle SCD dédiée.
Le tOracleSCDELT répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table Oracle SCD dédiée.

Propriétés du tOracleSCDELT Standard

Ces propriétés sont utilisées pour configurer le tOracleSCDELT s’exécutant dans le framework de Jobs Standard.
Le composant tOracleSCDELT Standard appartient aux familles Business Intelligence et Databases.
Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existante entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

| Connection type | Liste déroulante des pilotes disponibles:  
*Oracle OCI* : Sélectionnez ce type de connexion pour utiliser l'interface d'appel de la base de données Oracle (*Oracle Call Interface*) accompagnée d'un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle.  
*Oracle Service Name* : Sélectionnez ce type de connexion pour utiliser l'alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante.  
*Oracle SID* : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.  
*Oracle Custom* : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version d'Oracle que vous utilisez.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d'authentification de l'utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ *Password*, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur *OK* afin de sauvegarder les paramètres. |
| Source table | Nom de la table contenant les données à filtrer. |
| Table | Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois pour que l'opération d'insert soit autorisée. |
| Action on table | Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :  
*None* : n'effectuer aucune opération de table.  
*Drop and create the table* : supprimer la table puis en créer une nouvelle.  
*Create a table* : créer une table qui n'existe pas encore.  
*Create table if doesn’t exist* : créer la table si nécessaire. |
<table>
<thead>
<tr>
<th>Drop a table if exists and create</th>
<th>supprimer la table si elle existe déjà, puis en créer une nouvelle.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear a table</td>
<td>supprimer le contenu de la table.</td>
</tr>
<tr>
<td>Truncate table</td>
<td>supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.</td>
</tr>
</tbody>
</table>

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Built-in

Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository


### Surrogate Key

 Sélectionnez dans la liste une colonne à utiliser comme clé de substitution.

### Creation

 Sélectionnez la méthode à utiliser pour générer la clé de substitution.

Pour plus d’informations concernant les méthodes de création, consultez Méthodologie de gestion du SCD à la page 2716.

### Source Keys

 Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l’unicité des données entrantes.

### Source fields value include Null

 Cochez cette case pour autoriser les valeurs Nulls dans les colonnes source.

<i>Remarque :</i>

Les colonnes source font référence aux champs définis dans les tables SCD type 1 fields et SCD type 2 fields.
**Use SCD Type 1 fields**

Utilisez le type 1 si vous n’avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.

**Use SCD Type 2 fields**

Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.

- **Start date**: Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l’une des colonnes d’entrée du schéma comme date de départ (Start Date) dans la table SCD.
- **End Date**: Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne **End Date** a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l’option **Fixed Year value** et saisir une année fictive.
- **Log Active Status**: Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut `true` et `false`. Cette colonne permet de repérer facilement le journal actif.
- **Log versions**: Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.

**Advanced settings**

| Additional JDBC parameters | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n’est pas disponible lorsque vous utilisez l’option **Use an existing connection** dans les Basic settings. |
| Debug mode | Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |

2969
Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type <strong>Row Main</strong>.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.  

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**. |

### Scénario associé

Pour un scénario associé, consultez **tOracleSCD** à la page 2961, ainsi que **tMysqlSCD** à la page 2712.
tOracleSP

Ce composant appelle une procédure stockée de base de données Oracle.

Le tOracleSP permet de centraliser des requêtes multiples ou complexes dans une base de données et de les appeler plus facilement.

Propriétés du tOracleSP Standard

Ces propriétés sont utilisées pour configurer le tOracleSP s’exécutant dans le framework de Jobs Standard.

Le composant tOracleSP Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà défini.</td>
</tr>
<tr>
<td>Connection type</td>
<td>Liste déroulante des pilotes disponibles:</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.
Oracle OCI : Sélectionnez ce type de connexion pour utiliser l’interface d’appel de la base de données Oracle (Oracle Call Interface) accompagnée d’un ensemble de logiciels API de langage C qui fournissent une interface à cette base de données Oracle.

Oracle Service Name : Sélectionnez ce type de connexion pour utiliser l’alias TNS que vous fournissez lorsque vous vous connectez à la base de données distante.

Oracle SID : Sélectionnez ce type de connexion pour identifier exclusivement une base de données spécifique sur un système.

Oracle Custom : Sélectionnez ce type de connexion pour accéder à une base de données contenant des clusters.

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier des propriétés du composant. Les champs suivants sont alors pré-remplis.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version d’Oracle que vous utilisez.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification sur l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Dans une procédure stockée, le schéma est un paramètre d’entrée.</td>
</tr>
<tr>
<td></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma soit local (built-in) soit distant dans le Repository.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
</tbody>
</table>
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

**Repository** : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il est réutilisable. Voir également le **Guide utilisateur du Studio Talend**.

**SP Name**
Saisissez le nom exact de la Procédure Stockée (SP).

**Is Function / Return result in**
Cochez cette case, si une seule valeur doit être retournée.
Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.

**Parameters**
Cliquez sur le bouton `[+]` et sélectionnez dans le champ **Schema Columns** les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu’il n’y a de paramètres utilisés dans la procédure.

Sélectionnez le **Type** de paramètre :
- **IN** : paramètre d’entrée (Input).
- **OUT** : paramètre de sortie (Output)/valeur retournée.
- **IN OUT** : les paramètres d’entrée doivent être retournées sous forme de valeur, même après modifications via la procédure (fonction).
- **RECORDSET** : les paramètres d’entrée doivent être retournées sous forme d’ensemble de valeurs, au lieu d’une valeur unique.

**Remarque** :
Consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620 si vous voulez analyser un ensemble d’enregistrements d’une table de données ou d’une requête SQL.

Le type **Custom Type** est utilisé quand une colonne du schéma (**Schema Column**) que vous souhaitez utiliser est définie par l’utilisateur. Deux types de Custom sont disponibles dans le tableau **Parameters**.

Pour le premier type :
- Cochez la case quand la colonne du schéma correspondant est définie par l’utilisateur.
- Si toutes les colonnes du schéma dans le tableau **Parameters** sont de type personnalisé, vous pouvez cocher la case devant **Custom Type**.

<table>
<thead>
<tr>
<th>Sélectionnez un type de base de données dans la liste <strong>DB Type</strong> afin de mapper le type de la base de données source vers le type de la base de données cible :</th>
</tr>
</thead>
<tbody>
<tr>
<td>- <strong>Auto-Mapping</strong> : Mappe le type de la base de données source au type de la base de données cible automatiquement (par défaut).</td>
</tr>
<tr>
<td>- <strong>CLOB</strong> : Character large object.</td>
</tr>
<tr>
<td>- <strong>BLOB</strong> : Binary large object.</td>
</tr>
<tr>
<td>- <strong>DECIMAL</strong> : objet décimal numérique.</td>
</tr>
<tr>
<td>- <strong>NUMERIC</strong> : caractère compris entre 0 et 9.</td>
</tr>
<tr>
<td>- <strong>XMLTYPE</strong> : type du schéma XML.</td>
</tr>
</tbody>
</table>

⚠️ **Avertissement** :
Lorsque vous mappez une colonne sélectionnée dans la liste **Return result in** ou configurée comme type IN ou OUT du paramètre XMLTYPE, assurez-vous que la colonne est de type String dans le schéma.

Dans la seconde colonne **Custom Type**, vous pouvez préciser le type de personnalisation. Il peut être :
- **STRUCT** : utilisé pour un élément.
- **ARRAY** : utilisé pour plusieurs éléments.

Dans la colonne **Custom name**, spécifiez le type de personnalisation.

⚠️ **Avertissement** :
Dans le tableau **Parameters**, si une colonne est de **Type OUT**, assurez-vous que son type dans le schéma est bien **Object**.

**Specify a data source alias**
Cochez cette case et spécifiez l’alias de la source de données créée dans **Talend Runtime** pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans **Talend Runtime**. Pour voir un exemple de cas d’utilisation, consultez **Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL** à la page 2647.

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>Additional JDBC parameters</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case <strong>Use an existing connection</strong> est décochée dans les <strong>Basic settings</strong>.</td>
</tr>
</tbody>
</table>
### NLS Language
A partir de la liste déroulante, définissez la langue utilisée pour les données qui ne sont pas codées en Unicode.

### NLS Territory
Cochez cette case pour modifier les conventions d’appellation utilisées pour la numérotation des jours et des semaines. La valeur par défaut est celle du système d’exploitation.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau du composant.

### Utilisation

#### Règle d’utilisation
Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d’entrée sont autorisés.

#### Dynamic settings
Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


#### Limitation
La syntaxe de la Procédure Stockée doit correspondre à celle de la base de données. Quand le paramètre défini dans ce composant est personnalisé (Custom Type), les composants de la famille Java doivent être placés avant le tOracleSP, afin que les utilisateurs puissent définir des valeurs pour les paramètres personnalisés, ou bien après le tOracleSP, pour lire et écrire les paramètres personnalisés de type OUT.
Scénario : Vérifier le format de numéros à l’aide d’une procédure stockée

Le Job suivant permet de se connecter à une base Oracle contenant des numéros de sécurité social et le nom de leur propriétaire, en appelant une procédure stockée qui vérifie le format des numéros en fonction du format standard ###-##-####. Puis le résultat de la vérification, 1 pour un format valide et 0 pour un format non valide, est affiché en sortie dans la console de la vue Run.

- Cliquez et déposez les composants suivants : tOracleConnection, tOracleInput, tOracleSP et tLogRow.
- Reliez le composant tOracleConnection au tOracleInput à l’aide d’un lien Then Run puisqu’aucune donnée n’est traitée ici.
- Connectez les autres composants à l’aide de liens Row Main puisque des lignes sont transmises comme paramètres au composant SP et à la console.
- Dans l’onglet Basic settings du composant tOracleConnection, paramétrez les informations de connexion à la base de données correspondante. Vous pourrez donc réutiliser cette information dans tout autre composant lié aux bases de données.
- Puis double-cliquez sur le composant tOracleInput et paramétrez ses propriétés.
  - Cochez la case Use an existing connection et sélectionnez le composant tOracleConnection dans la liste pour réutiliser les informations de connexion déjà paramétrées.
  - Sélectionnez l’option Repository dans le champ Property type puisque le schéma Oracle est défini dans une métadonnée du dossier Db Connections du Repository. Si vous n’avez pas enregistré de métadonnée contenant les informations de connexion à la base de données Oracle dans le Repository, renseignez le nom du schéma manuellement.
• Puis sélectionnez l’option **Repository** dans le champ **Schema**, et récupérez le schéma correspondant à la table de la base de données Oracle.

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>CITY</th>
<th>SSNUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jack</td>
<td>LA</td>
<td>123-45-6789</td>
</tr>
<tr>
<td>2</td>
<td>Tom</td>
<td>NYC</td>
<td>123-45-6789</td>
</tr>
<tr>
<td>3</td>
<td>Bill</td>
<td>SF</td>
<td>123-45-6789</td>
</tr>
<tr>
<td>4</td>
<td>Jana</td>
<td>NYC</td>
<td>236-52-2956</td>
</tr>
<tr>
<td>5</td>
<td>Brandon</td>
<td>SLC</td>
<td>561-52-B267</td>
</tr>
</tbody>
</table>

• Dans cet exemple, la table contenant le numéro de sécurité social contient un schéma à quatre colonnes dont : **ID, NAME, CITY et SSNUMBER**.

• Dans le champ **Query**, saisissez la requête Select suivante ou sélectionnez-la dans la liste si vous l’avez stockée dans le Repository.

```sql
select ID, NAME, CITY, SSNUMBER from SSN
```

• Puis sélectionnez le **tOracleSP** et paramétrez ses propriétés dans l’onglet **Basic settings**.

- **Use an existing connection**
- **Component list**
- **tOracleConnection**
- **Schema**
- **Repository**
- **DB (ORACLE_SERVICE_NAME):oracleconnection**
- **Edit schema**
- **Sync columns**
- **SP Name**
- **"s_ssn"**
- **Return result in**
- **SSN_Verified**
- **DBType**
- **AUTO-MAPPING**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Scheme Column</th>
<th>Type</th>
<th>DBType</th>
<th>Nullable</th>
<th>Default Type</th>
<th>Custom Type</th>
<th>Custom Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSNUMBER</td>
<td>IN</td>
<td>AUTO-MAPPING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Comme pour le composant **tOracleInput**, sélectionnez l’option **Repository** dans le champ **Property type** et cochez la case **Use an existing connection**, puis sélectionnez l’entrée adéquate dans la liste **Component List**.

• Le schéma utilisé pour le composant **tOracleSP** diffère légèrement du schéma d’entrée. En effet, une colonne supplémentaire (**SSN_Valid**) est ajoutée au schéma d’entrée. Cette colonne contiendra le statut de validité (**1 ou 0**) généré par la procédure.

• Puis sélectionnez le type d’encodage adéquat dans la liste **Encoding type**.
Dans le champ **SP Name**, saisissez le nom exact de la procédure stockée (ou de la fonction) telle qu’elle est appelée dans la base de données. Dans cet exemple, le nom de la procédure stockée est **is_ssn**.

La fonction de base utilisée pour cet exemple est :

```
CREATE OR REPLACE FUNCTION is_ssn(string_in VARCHAR2)
RETURN PLS_INTEGER
IS
    -- validating ###-##-#### format
BEGIN
    IF TRANSLATE(string_in, '0123456789A', 'AAAAAAAAAAB') = 'AAA-AA-AAAA' THEN
        RETURN 1;
    END IF;
    RETURN 0;
END is_ssn;
/
```

Comme un valeur retournée est attendue dans cet exemple, la procédure agit comme une fonction, ainsi cochez la case **Is function**.

La seule valeur retournée attendue est basée sur la colonne **ssn_valid**. Ainsi, sélectionnez l’entrée correspondante dans la liste.

Dans la zone **Parameters**, définissez les paramètres d’entrée et de sortie utilisés dans la procédure. Dans cet exemple, seule la colonne **SSNumber** du schéma est utilisée dans la procédure.

Cliquez sur le signe [+] pour ajouter un nouvelle ligne à la table et sélectionnez la colonne adéquate (**SSNumber**) et son type (**IN**).

Puis sélectionnez le composant **tLogRow** et cliquez sur **Sync Column** pour récupérer le schéma du composant précédent (**tOracleSP**).

Cochez la case **Print values in cells of a table** pour faciliter la lecture de la sortie.

Puis enregistrez le Job et appuyez sur **F6** pour l’exécuter.
Dans la console, vous pouvez lire les résultats en sortie. Toutes les colonnes du schéma d’entrée sont affichées même si elles ne sont pas utilisées comme paramètres dans la procédure stockée.

La colonne finale affiche la valeur retournée attendue, c’est-à-dire que le numéro de sécurité sociale vérifié soit valide ou non.

Consultez également Scénario : Insérer des données dans des tables mère/fille à la page 2620 si vous voulez analyser un ensemble d’enregistrements d’une table de données ou d’une requête SQL.

**Scénarios associés**

Pour des scénarios associés, consultez :

- **Scénario : Récupérer des informations personnelles à l’aide d’une procédure stockée** à la page 2594.
- **Scénario : Utiliser le tMysqlSP pour trouver le libellé State à l’aide d’une procédure stockée** à la page 2734.
- **Scénario : Exécuter une procédure stockée à l’aide du tMDMSP** à la page 2320.
tOracleTableList

Ce composant liste les noms des tables Oracle à l’aide d’une instruction SELECT se basant sur une clause WHERE.

Le tOracleTableList effectue une opération d’itération sur toutes les tables d’une base de données, grâce à une connexion Oracle définie.

Propriétés du tOracleTableList Standard

Ces propriétés sont utilisées pour configurer le tOracleTableList s’exécutant dans le framework de Jobs Standard.

Le composant tOracleTableList Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Component list</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez le composant de connexion tOracleConnection dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Where clause for table name selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisissez la commande WHERE permettant d’identifier les tables sur lesquelles effectuer l’opération d’itération.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez la case afin de collecter les données de log au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURRENT_TABLE : nom de la table sur laquelle se fait l’itération. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>NB_TABLE : nombre de tables itérées jusqu’à présent. Cette variable est une variable Flow et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est</td>
</tr>
</tbody>
</table>
une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Il faut utiliser ce composant en association avec les autres composants Oracle, notamment avec le tOracleConnection.</th>
</tr>
</thead>
</table>
| **Dynamic settings** | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

### Scénario associé

Pour un scénario associé, consultez Scénario : Effectuer une opération d’itération sur des tables et en effacer le contenu grâce à un modèle SQL défini par l’utilisateur (SQL Template) à la page 2741.
tPaloCheckElements

Ce composant vérifie l’existence d’éléments du flux de données d’entrée dans un cube donné.

Le tPaloCheckElements peut être utilisé avec le tPaloOutputMulti. Il vérifie si les éléments du flux d’entrée existent dans le cube donné, avant de les écrire. Il peut également définir une valeur par défaut qui peut être utilisée si les éléments n’existent pas.

Propriétés du tPaloCheckElements Standard

Ces propriétés sont utilisées pour configurer le tPaloCheckElements s’exécutant dans le framework de Jobs Standard.

Le composant tPaloCheckElements Standard appartient à la famille Business Intelligence.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.</td>
</tr>
<tr>
<td>Host Name</td>
<td>Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte.</td>
</tr>
<tr>
<td>Server Port</td>
<td>Saisissez le numéro du port d’écoute du serveur Palo.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données dans laquelle écrire les données.</td>
</tr>
<tr>
<td>Cube</td>
<td>Saisissez le nom du cube dans lequel les données doivent être écrites.</td>
</tr>
<tr>
<td>On element error</td>
<td>Sélectionnez l’action à effectuer lorsqu’un élément n’existe pas. - Reject row : la ligne correspondante sera rejetée et placée dans le flux de rejet. - Use default : la valeur par défaut définie sera utilisée. - Stop : le processus complet sera interrompu.</td>
</tr>
</tbody>
</table>
### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

#### Built-in

Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

#### Repository


### Advanced settings

Tête de Colonnes : affiche la (les) colonne(s) du schéma d’entrée. Cette colonne est automatiquement remplie une fois qu’un schéma d’entrée est saisi ou créé.

- **Element type** : sélectionnez le type d’élément de la colonne d’entrée. Seule une colonne peut être définie en tant que **Measure**.

- **Default** : saisissez la valeur par défaut qui sera utilisée si l’option **Use default** a été sélectionnée dans le champ **On element error**.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</th>
</tr>
</thead>
</table>

Cochez cette case pour collecter les données de log au niveau du composant.
Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant requiert un composant d’entrée.</th>
</tr>
</thead>
</table>

**Connections**

Liens de sortie (de ce composant à un autre) :
- **Row** : Main, Rejects
- **Trigger** : Run if, On Component Ok, On Component Error.

Liens d’entrée (d’un autre composant à celui-ci) :
- **Row** : Main, Rejects

Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.

**Limitation**

Ce composant fonctionne uniquement avec des cubes Palo de type “Normal”.


**Scénario associé**

Pour un scénario associé, consultez Scénario 2 : Refuser un flux de données d’entrée lorsque les éléments à écrire n’existent pas dans un cube donné à la page 3035.
tPaloClose

Ce composant ferme une connexion active au serveur Palo.
Le tPaloClose ferme une connexion à un serveur Palo afin de libérer des ressources occupées.

Propriétés du tPaloClose Standard

Ces propriétés sont utilisées pour configurer le tPaloClose s'exécutant dans le framework de Jobs Standard.
Le composant tPaloClose Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.

Basic settings

| Component List | Sélectionnez un composant tPaloConnection dans la liste déroulante si plusieurs connexions sont utilisées dans le Job. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case afin de collecter les données au niveau du composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Palo, notamment avec le tPaloConnection. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de |
contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

Lorsqu'un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Scénario associé

Aucun scénario n'est disponible pour la version Standard de ce composant.
tPaloConnection

Ce composant ouvre une connexion à un serveur Palo et permet à d’autres composants Palo de partager cette connexion durant le processus complet.

Propriétés du tPaloConnection Standard

Ces propriétés sont utilisées pour configurer le tPaloConnection s’exécutant dans le framework de Jobs Standard.

Le composant tPaloConnection Standard appartient à la famille Business Intelligence.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Host Name</th>
<th>Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Port</td>
<td>Saisissez le numéro du port d’écoute du serveur Palo.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de Palo.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics       | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables


2987
**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est utilisé avec les composants Palo afin de partager une connexion à un serveur Palo.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connections</td>
<td></td>
</tr>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez **Scénario : Créer une dimension avec des éléments** à la page 3010.
tPaloCube

Ce composant effectue des opérations sur un cube Palo donné.
Le tPaloCube crée, supprime ou vide des cubes Palo à partir de dimensions existantes dans une base de données Palo.

Propriétés du tPaloCube Standard

Ces propriétés sont utilisées pour configurer le tPaloCube s'exécutant dans le framework de Jobs Standard.
Le composant tPaloCube Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.

Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
| Host Name | Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte. |
| Server Port | Saisissez le numéro du port d’écoute du serveur Palo. |
| Username et Password | Informations d’authentification de l’utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Database | Saisissez le nom de la base de données dans laquelle l’opération donnée doit s’effectuer. |
| Cube | Saisissez le nom du cube dans lequel l’opération donnée doit s’effectuer. |
| Cube type | Sélectionnez le type de cube dans la liste déroulante pour le cube sur lequel l’opération donnée doit s’effectuer :
- Normal : Type de cube "normal", type par défaut.
- Attribut : un cube Attribute sera créé avec la création d’un cube "normal".
- User Info : un cube User Info sera créé/modifié par ce composant. |
<table>
<thead>
<tr>
<th>Action on cube</th>
<th>Sélectionnez l’opération que vous souhaitez effectuer sur le cube défini :</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Create cube</td>
<td>le cube n’existe pas et sera créé.</td>
</tr>
<tr>
<td>- Create cube if not exists</td>
<td>crée un cube s’il n’existe pas.</td>
</tr>
<tr>
<td>- Delete cube if exists and create</td>
<td>supprime un cube s’il existe et en crée un nouveau.</td>
</tr>
<tr>
<td>- Delete cube</td>
<td>supprime le cube de la base de données.</td>
</tr>
<tr>
<td>- Clear cube</td>
<td>supprime les données du cube.</td>
</tr>
</tbody>
</table>

| Dimension list | Ajoutez des lignes en cliquant sur le bouton [+] et dans chaque ligne, saisissez le nom d’une dimension existant dans la base de données et devant être utilisée dans le cube. L’ordre de cette liste définit également l’ordre de création des dimensions. |

<table>
<thead>
<tr>
<th>Advanced settings</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Global Variables</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Variables</td>
<td><strong>CUBENAME</strong> : nom du cube. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Utilisation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Règle d’utilisation</td>
<td>Ce composant peut être utilisé en standalone dans la création dynamique de cubes, avec une liste définie de dimensions.</td>
</tr>
<tr>
<td>Connections</td>
<td>Liens de sortie (de ce composant à un autre) :</td>
</tr>
<tr>
<td></td>
<td>Liens d’entrée (d’un autre composant à celui-ci) :</td>
</tr>
<tr>
<td><strong>tPaloCube</strong></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td><strong>Row</strong> : Iterate</td>
<td></td>
</tr>
<tr>
<td>Pour plus d’informations concernant les connexions, consultez la section relative aux différents types de connexions, dans le <em>Guide utilisateur du Studio Talend</em>.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Limitation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Le cube créant un processus ne crée pas les dimensions à partir de rien. Les dimensions utilisées dans le cube ont été créées précédemment.</td>
</tr>
<tr>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

**Scénario : Créer un cube dans une base de données existante**

Le Job de ce scénario crée un nouveau cube à deux dimensions dans la base de données démo *Biker* de Palo.

Pour reproduire ce scénario, procédez comme suit :

**Configurer le composant tPaloCube**

**Procédure**

1. Déposez un composant tPaloCube de la Palette dans l’espace de modélisation graphique.
2. Double-cliquez sur le composant tPaloCube afin d’ouvrir sa vue Component.
3. Dans le champ Host name, saisissez le nom de l’hôte ou l’adresse IP du serveur hôte, localhost dans cet exemple.

4. Dans le champ Server Port, saisissez le numéro du port d’écoute du serveur Palo. Dans ce scénario, le numéro est 7777.

5. Dans les champs Username (identifiant) et Password (mot de passe), saisissez vos informations d’authentification. Dans cet exemple, les deux sont admin.

6. Dans le champ Database, saisissez le nom de la base de données dans laquelle vous souhaitez créer le cube, Biker dans cet exemple.

7. Dans le champ Cube, saisissez le nom que vous souhaitez utiliser afin de créer le cube, bikerTalend, par exemple.

8. Dans le champ Cube type, sélectionnez le type Normal dans la liste déroulante pour le cube à créer, ce qui signifie que ce cube sera de type “Normal”, le type par défaut.

9. Dans le champ Action on cube, sélectionnez l’action à effectuer. Dans ce scénario, sélectionnez Create cube.

10. Sous le tableau Dimensionlist, cliquez deux fois sur le bouton [+] afin d’ajouter deux lignes.

11. Dans le tableau Dimensionlist, saisissez le nom de chaque nouvelle ligne afin de remplacer le nom par défaut. Dans ce scénario, saisissez Months pour la première ligne et Products pour la seconde. Ces deux dimensions existent déjà dans la base de données Biker dans laquelle le nouveau cube sera créé.

**Exécuter le Job**

Appuyez sur F6 afin d’exécuter le Job.

Un nouveau cube a été créé dans la base de données Biker et les deux dimensions sont ajoutées dans ce cube.
tPaloCubeList

Ce composant récupère les informations d'une liste de cubes dans la base de données Palo déterminée.

Le tPaloCubeList liste le nom des cubes, leur types, le nombre de dimensions assignées, le nombre de cellules remplies dans la base de données déterminée.

Découvrir le schéma de sortie en lecture seule du tPaloCubeList

Le tableau ci-dessous présente les informations relatives au schéma en lecture seule du composant tPaloCubeList.

<table>
<thead>
<tr>
<th>Colonne</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cube_id</td>
<td>int</td>
<td>ID interne du cube.</td>
</tr>
<tr>
<td>Cube_name</td>
<td>string</td>
<td>Nom du cube.</td>
</tr>
<tr>
<td>Cube_dimensions</td>
<td>int</td>
<td>Nombre de dimensions dans le cube.</td>
</tr>
<tr>
<td>Cube_cells</td>
<td>long</td>
<td>Nombre de cellules à l'intérieur du cube.</td>
</tr>
<tr>
<td>Cube_filled_cells</td>
<td>long</td>
<td>Nombre de cellules remplies dans le cube.</td>
</tr>
<tr>
<td>Cube_status</td>
<td>int</td>
<td>Statut du cube. Il peut être :</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0 : non chargé</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 : chargé</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2 : changé</td>
</tr>
<tr>
<td>Cube_type</td>
<td>int</td>
<td>Type du cube. Il peut être :</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0 : normal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 : system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2 : attribute</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 3 : user info</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 4 : gpu type</td>
</tr>
</tbody>
</table>

Propriétés du tPaloCubeList Standard

Ces propriétés sont utilisées pour configurer le tPaloCubeList s'exécutant dans le framework de Jobs Standard.

Le composant tPaloCubeList Standard appartient à la famille Business Intelligence.

Le composant de ce framework est toujours disponible.
### Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste **Component List** présente uniquement les composants de connexion du Job du même niveau. |
| Host Name | Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte. |
| Server Port | Saisissez le numéro du port d’écoute du serveur Palo. |
| Username et Password | Informations d’authentification de l’utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| Database | Saisissez le nom de la base de données contenant les cubes dont vous souhaitez récupérer les informations. |

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | **NB_CUBES** : nombre de cubes. Cette variable est une variable **After** et retourne un nombre entier.  
**CUBEID** : ID du cube. Cette variable est une variable **Flow** et retourne un nombre entier.  
**CUBENAME** : nom du cube. Cette variable est une variable **Flow** et retourne une chaîne de caractères.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en tant que composant de début. Il requiert un composant de sortie.</th>
</tr>
</thead>
</table>
| Connections         | Liens de sortie (de ce composant à un autre) :
|                     | Liens d’entrée (d’un autre composant à celui-ci) :
|                     | Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend. |

### Limitation


### Scénario : Récupérer des informations détaillées d’un cube d’une base de données déterminée

Le Job dans ce scénario récupère des informations détaillées concernant les cubes de la base de données démo de Palo Biker.

Pour reproduire ce scénario, procédez comme suit :

#### Configurer le Job

**Procédure**

1. Déposez un composant **tPaloCubeList** et un **tLogRow** de la Palette dans l’espace de modélisation graphique.
2. Cliquez-droit sur le tPaloCubeList afin d’ouvrir son menu contextuel.
3. Dans ce menu, sélectionnez Row > Main pour relier les deux composants.

**Configurer le composant tPaloCube**

**Procédure**

1. Double-cliquez sur le tPaloCube afin d’ouvrir sa vue Component.

![Component settings window](image)

3. Dans le champ **Server Port**, saisissez le numéro du port d’écoute du serveur Palo. Dans ce scénario, le numéro est 7777.
4. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d’authentification. Dans cet exemple, les deux sont *admin*.

**Exécuter le Job**

Appuyez sur F6 pour exécuter le Job.

Les informations du cube sont récupérées de la base de données *Biker* et sont affichées dans la console de la vue Run.

```
[statistics] connected
26|Orders|8|45291236350400|133005|1|0
27|bikerTalend|2|11830|1|0|1|0
1|GROUP_CUBE_DATA|2|0|0|1|1
2|CONFIGURATION|1|2|2|1|1
2|CONFIGURATION|1|2|2|1|1
6|SUBSET_LOCAL|3|0|0|1|1
6|SUBSET_LOCAL|3|0|0|1|1
7|SUBSET_GLOBAL|2|0|0|1|1
7|VIEW_STORE|3|0|0|1|1
8|VIEW_STORE|3|0|0|1|1
8|VIEW_GLOBAL|2|0|0|1|1
9|VIEW_GLOBAL|2|0|0|1|1
11|GROUP_DIMENSION_DATA_Products|2|1352|0|1|1
13|GROUP_DIMENSION_DATA_Customers|2|1864|0|1|1
```
Pour plus d’informations concernant l’interprétation des informations sur le cube, consultez la Découvrir le schéma de sortie en lecture seule du tPaloCubeList à la page 2994.
tPaloDatabase

Ce composant gère les bases de données dans un serveur Palo.
Le tPaloDatabase crée, supprime ou recrée des bases de données sur un serveur Palo donné.

Propriétés du tPaloDatabase Standard

Ces propriétés sont utilisées pour configurer le tPaloDatabase s’exécutant dans le framework de Jobs Standard.
Le composant tPaloDatabase Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.

Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.
Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
| Host Name | Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte. |
| Server Port | Saisissez le numéro du port d’écoute du serveur Palo. |
| Username et Password | Informations d’authentification de l’utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Database | Saisissez le nom de la base de données dans laquelle l’opération doit s’effectuer. |
| Action on database | Sélectionnez l’opération que vous souhaitez effectuer sur le base de données:
- **Create database** : la base de données n’existe pas et sera créée.
- **Create database if not exists** : la base est créée lorsqu’elle n’existe pas.
- **Delete database if exists and create** : la base de données est supprimée si elle existe et une nouvelle base de données est créée.
- **Delete database** : la base de données est supprimée du serveur. |
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | DATABASENAME : nom de la base de données. Cette variable est une variable After et retourne une chaîne de caractères.  
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.  
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en standalone lors d’opérations de gestion de bases de données dans un serveur Palo. |

| Connections | Liens de sortie (de ce composant à un autre) :  
Liens d’entrée (d’un autre composant à celui-ci) :  
Row : Iterate  
Trigger : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error  
Pour plus d’informations concernant les connexions, consultez la section relative aux différents types de connexions dans le Guide utilisateur du Studio Talend. |

| Limitation | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton Install dans l’onglet Component. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet Modules de la perspective Integration de votre studio. Vous pouvez trouver plus d’informations concernant l’installation |
Scénario : Créer une base de données

Le Job de ce scénario crée une nouvelle base de données sur un serveur Palo donné.

Pour reproduire ce scénario, procédez comme suit :

**Procédure**

1. Déposez un composant **tPaloDatabase** de la Palette dans l’espace de modélisation graphique.
2. Double-cliquez sur le composant **tPaloDatabase** afin d’ouvrir sa vue **Component**.
3. Dans le champ **Host name**, saisissez le nom de l’hôte ou l’adresse IP du serveur hôte, **localhost** dans cet exemple.
4. Dans le champ **Server Port**, saisissez le numéro du port d’écoute du serveur Palo. Dans ce scénario, le numéro est 7777.
5. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d’authentification. Dans cet exemple, les deux sont **admin**.
6. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle vous souhaitez créer le cube, **talenddatabase** dans cet exemple.
7. Dans le champ **Action on database**, sélectionnez l’action à effectuer. Dans ce scénario, sélectionnez **Create database** puisque la base de données à créer n’existe pas.
8. Appuyez sur **F6** pour exécuter votre Job.
Résultats

Une nouvelle base de données est créée sur le serveur Palo donné.
tPaloDatabaseList

Ce composant liste les noms des bases de données, leur type, le nombre de cubes, le nombre de dimensions, le statut et l’ID des bases de données d’un serveur Palo donné.

Le tPaloDatabaseList récupère les informations d’une liste de bases de données dans un serveur Palo donné.

Découvrir le schéma de sortie en lecture seule du tPaloDatabaseList

Le tableau ci-dessous présente les informations relatives au schéma de sortie en lecture seule du composant tPaloDatabaseList.

<table>
<thead>
<tr>
<th>Base de données</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database_id</td>
<td>long</td>
<td>ID interne de la base de données.</td>
</tr>
<tr>
<td>Database_name</td>
<td>string</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Database_dimensions</td>
<td>int</td>
<td>Nombre de dimensions dans la base de données.</td>
</tr>
<tr>
<td>Database_cubes</td>
<td>int</td>
<td>Nombre de cubes dans la base du données.</td>
</tr>
<tr>
<td>Database_status</td>
<td>int</td>
<td>Statut de la base de données.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0 = non chargée</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 = chargée</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2 = changée</td>
</tr>
<tr>
<td>Database_types</td>
<td>int</td>
<td>Type de la base de données.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0 = normal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 = system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 3 = user info</td>
</tr>
</tbody>
</table>

Propriétés du tPaloDatabaseList Standard

Ces propriétés sont utilisées pour configurer le tPaloDatabaseList s’exécutant dans le framework de Jobs Standard.

Le composant tPaloDatabaseList Standard appartient à la famille Business Intelligence.

Le composant de ce framework est toujours disponible.

Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |
Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.

<table>
<thead>
<tr>
<th>Host Name</th>
<th>Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Port</td>
<td>Saisissez le numéro du port d’écoute du serveur Palo.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de Palo.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_DATABASES : nombre de bases de données. Cette variable est une variable After et retourne un nombre entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATABASEID : ID de la base de données. Cette variable est une variable Flow et retourne un long.</td>
</tr>
<tr>
<td></td>
<td>DATABASENAME : nom de la base de données. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

| Règle d’utilisation | Ce composant peut être utilisé en tant que composant de début. Il nécessite un composant de sortie. |
**Scénario : Récupérer des informations détaillées concernant les bases de données d’un serveur Palo donné**

Le Job de ce scénario récupère des informations concernant toutes les bases de données d’un serveur Palo donné.

Pour reproduire ce scénario, procédez comme suit :

**Configurer le Job**

**Procédure**

1. Déposez un composant tPaloDatabaseList et un tLogRow de la Palette dans l’espace de modélisation graphique.
2. Cliquez-droit sur le tPaloDatabaseList pour ouvrir le menu contextuel.
3. Dans ce menu, sélectionnez Row > Main pour relier les deux composants.
Configurer le composant tPaloDatabaseList

Procédure

1. Double-cliquez sur le tPaloDatabaseList pour ouvrir sa vue Component.

![tPaloDatabaseList_1](image)

2. Dans le champ Host name, saisissez le nom de l’hôte ou l’adresse IP du serveur hôte, localhost dans cet exemple.

3. Dans le champ Server Port, saisissez le numéro du port d’écoute du serveur Palo. Dans ce scénario, le numéro est 7777.

4. Dans les champs Username (identifiant) et Password (mot de passe), saisissez vos informations d’authentification. Dans cet exemple, les deux sont admin.

Exécuter le Job

Appuyez sur F6 pour exécuter le Job.

Les informations de toutes les bases de données du serveur Palo sont récupérées et affichées dans la console de la vue Run.

<table>
<thead>
<tr>
<th>0</th>
<th>System</th>
<th>7</th>
<th>6</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Demo</td>
<td>23</td>
<td>29</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Biker</td>
<td>27</td>
<td>28</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>tx_g</td>
<td>17</td>
<td>17</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>DBASE24</td>
<td>13</td>
<td>12</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>31</td>
<td>elaCreate</td>
<td>13</td>
<td>12</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>tPaloOutputMulti</td>
<td>15</td>
<td>15</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant le schéma de sortie, consultez Découvrir le schéma de sortie en lecture seule du tPaloDatabaseList à la page 3003.
tPaloDimension

Ce composant gère les dimensions Palo, ainsi que les éléments dans une base de données.
Le tPaloDimension crée, supprime ou recrée des dimensions avec ou sans éléments de dimension dans une base de données Palo.

Propriétés du tPaloDimension Standard

Ces propriétés sont utilisées pour configurer le tPaloDimension s’exécutant dans le framework de Jobs Standard.
Le composant tPaloDimension Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.

Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
| Host Name | Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte. |
| Server Port | Saisissez le numéro du port d’écoute du serveur Palo. |
| Username et Password | Informations d’authentification de l’utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Database | Saisissez le nom de la base de données dans laquelle les dimensions sont gérées. |
| Dimension | Saisissez le nom de la dimension sur laquelle l’opération donnée doit s’effectuer. |
| Action on dimension | Sélectionnez l’opération que vous souhaitez effectuer sur la dimension :  
- **None** : aucune action n’est effectuée sur la dimension.  
- **Create dimension** : la dimension n’existe pas et sera créée.  
- **Create dimension if not exists** : cette dimension est créée quand elle n’existe pas. |
- **Delete dimension if exists and create** : cette dimension est supprimée si elle existe et une nouvelle dimension est créée.
- **Delete dimension** : la dimension est supprimée de la base de données.

### Create dimension elements
Cochez cette case pour activer les champs de gestion des dimensions et créer des éléments de dimension, et créer cette dimension.

### Advanced settings

#### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

#### DIMENSIONNAME
nom de la dimension. Cette variable est une variable After et retourne une chaîne de caractères.

#### ERROR_MESSAGE
message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consulitez le Guide utilisateur du Studio Talend.

### Utilisation

Ce composant peut être utilisé en standalone ou comme composant de fin d’un processus.

#### Remarque :
Les champs ci-dessous sont disponibles uniquement quand la case **Create dimension elements** est cochée.

#### Dimension type

#### Remarque :
Disponible uniquement lorsque l’action à effectuer sur la dimension est **None**.

#### Commit size

#### Schema et Edit Schema
<table>
<thead>
<tr>
<th><strong>Consolidation type - None</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remarque :</strong></td>
<td>Cette option active les champs des paramètres correspondants devant être renseignés.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Consolidation type - Normal</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remarque :</strong></td>
<td>Cette option active les champs des paramètres correspondants devant être renseignés.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Consolidation type - Self-referenced</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remarque :</strong></td>
<td>Cette option active les champs des paramètres correspondants devant être renseignés.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Element’s type</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Creation mode</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Connections**

Liens de sortie (de ce composant à un autre) :


Liens d’entrée (d’un autre composant à celui-ci) :

- **Row** : Main, Iterate

**Trigger** : Run if, On Subjob Ok, On Subjob Error, On Component Ok, On Component Error

Pour plus d’informations concernant les connexions, consultez la section relatives aux différents types de connexions, dans le *Guide utilisateur du Studio Talend*.

**Limitation**

La suppression d’éléments de dimension est possible uniquement avec le type de consolidation **None**. Seul
Scénario : Créer une dimension avec des éléments

Le Job de ce scénario crée une dimension de date avec une hiérarchie d’éléments simple, composée de trois niveaux : Year, Month, Date (Année, mois, date).

Pour reproduire ce scénario, procédez comme suit :

**Construire le Job**

**Procédure**

1. Déposez un composant **tPaloConnection**, un **tRowGenerator**, un **tMap**, et un **tPaloDimension** de la Palette dans l’espace de modélisation graphique.
2. Cliquez-droit sur le **tPaloConnection** afin d’ouvrir le menu contextuel et sélectionnez **Trigger > On Subjob Ok** pour le relier au **tRowGenerator**.
3. Cliquez-droit sur le **tRowGenerator** afin d’ouvrir le menu contextuel et sélectionnez **Row > Main** pour le relier au **tMap**.

 REMARQUE :

Le **tRowGenerator** est utilisé pour générer des lignes au hasard afin de simplifier ce processus. Dans un cas réel d’utilisation, vous pouvez utiliser un autre composant d’entrée afin de charger vos données.

4. Cliquez-droit sur le **tMap** pour ouvrir le menu contextuel et sélectionnez **Row > *New output*** pour le relier au **tPaloDimension**. Nommez ce lien ou1 dans la boîte de dialogue qui s’ouvre.
Paramétrer la connexion à une base de données

Procédure
1. Double-cliquez sur le composant **tPaloCube** afin d’ouvrir sa vue **Component**.

![tPaloConnection_1](image)

3. Dans le champ **Server Port**, saisissez le numéro du port d’écoute du serveur Palo. Dans ce scénario, le numéro est *7777*.
4. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez les informations d’authentification. Dans cet exemple, les deux sont *admin*.

Configurer le composant **Input**

Procédure
1. Double-cliquez sur le **tRowGenerator** pour ouvrir son éditeur.

2. Dans la partie supérieure de l’éditeur, cliquez sur le bouton `[+]` pour ajouter une colonne, et renommez-la **random_date** dans la colonne **Column**.
3. Dans la ligne nouvellement ajoutée, sélectionnez **Date** pour la colonne **Type**, et **getRandomDate** dans la colonne **Functions**.
4. Dans la vue **Function parameters** de la partie inférieure de l’éditeur, saisissez les nouvelles valeurs des dates minimum et maximum de la colonne **Value**. Dans cet exemple, le minimum est 2010-01-01, le maximum est 2010-12-31.

5. Cliquez sur **OK** afin de valider vos modifications et fermer l’éditeur.

6. Dans la boîte de dialogue qui s’ouvre, cliquez sur **OK** pour propager les modifications.

**Configurer le tMap editor**

**Procédure**

1. Double-cliquez sur le **tMap** afin d’ouvrir son éditeur.

2. Dans la vue **Schema editor** de la partie inférieure de la fenêtre, sous la table **out1**, cliquez sur le bouton "+" afin d’ajouter trois lignes.

3. Dans la colonne **Column** de la table **out1** saisissez de nouveaux noms pour les trois lignes ajoutées, **Year**, **Month** et **Date** (année, mois et date). Ces lignes sont ajoutées automatiquement à la table **out1** dans la partie inférieure du **tMap editor**.

4. Dans la table **out1** de la partie supérieure de l’éditeur, cliquez sur la colonne **Expression** de la ligne **Year**.

5. Appuyez sur les touches **Ctrl+Espace** afin d’ouvrir la liste d’auto-complétion.

6. Double-cliquez sur **TalendDate.formatDate** pour la sélectionner. L’expression pour obtenir la date s’affiche dans la ligne **Year**. L’expression est la suivante :

   ```
   TalendDate.formatDate("yyyy-MM-dd HH:mm:ss",myDate).
   ```

7. Remplacez l’expression par défaut par **TalendDate.formatDate("yyyy",row1.random_date)**.

8. Répétez l’opération pour les lignes **Month** et **Date** pour ajouter l’expression par défaut et replacez-la respectivement par **TalendDate.formatDate("MM",row1.random_date)** et par **TalendDate.formatDate("dd-MM-yyyy", row1.random_date)**.
Cliquez sur OK afin de valider cette modification et acceptez la propagation des modifications en cliquant sur OK dans la boîte de dialogue qui s’ouvre.

**Configurer le composant tPaloDimension**

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le tPaloDimension afin d’ouvrir sa vue Component.

2. Cochez la case Use an existing connection. Le tPaloConnection_1 s’affiche automatiquement dans le champ Connection configuration.

3. Dans le champ Database, saisissez le nom de la base de données dans laquelle la nouvelle dimension est créée, talendDatabase dans cet exemple.

4. Dans le champ Dimension, saisissez le nom que vous souhaitez utiliser pour la dimension à créer, par exemple.

5. Dans le champ Action on dimension, sélectionnez l’action à effectuer. Dans ce scénario, sélectionnez Create dimension if not exist.

6. Sélectionnez la case Create dimension elements.

7. Dans la zone Consolidation Type, sélectionnez le bouton Normal.

8. Sous le tableau de la hiérarchie des éléments, cliquez trois fois sur le bouton [+] pour ajouter trois lignes au tableau.

9. Dans la colonne Input column du tableau, sélectionnez Year dans la liste déroulante de la première ligne, Month dans la deuxième et Date dans la troisième. Cela détermine le niveau des éléments des différentes colonnes du schéma d’entrée.
**Exécuter le Job**

Appuyez sur **F6** pour exécuter le Job.

Une nouvelle dimension a été créée dans la base de données Palo *talendDatabase*.
tPaloDimensionList

Ce composant récupère les informations d'une liste de dimensions de la base de données Palo déterminée.
Le tPaloDimensionList affiche le nom des dimensions, leur type, le nombre d'éléments de dimension, le maximum d'indentation des dimensions, le maximum de profondeur des dimensions, le maximum des niveaux des dimensions, l'ID des dimensions dans un serveur Palo donné.

Découvrir le schéma de sortie en lecture seule du tPaloDimensionList

Le tableau ci-dessous présente les informations relatives au schéma de sortie en lecture seule du composant tPaloDimensionList.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension_id</td>
<td>long</td>
<td>ID interne de la dimension.</td>
</tr>
<tr>
<td>Dimension_name</td>
<td>string</td>
<td>Nom de la dimension.</td>
</tr>
<tr>
<td>Dimension_attribute_cube</td>
<td>string</td>
<td>Nom du cube des attributs.</td>
</tr>
<tr>
<td>Dimension_rights_cube</td>
<td>string</td>
<td>Nom du cube des droits.</td>
</tr>
<tr>
<td>Dimension_elements</td>
<td>int</td>
<td>Nombre d'éléments de dimension.</td>
</tr>
<tr>
<td>Dimension_max_level</td>
<td>int</td>
<td>Niveau maximal de la dimension.</td>
</tr>
<tr>
<td>Dimension_max_indent</td>
<td>int</td>
<td>Indentation maximum de la dimension.</td>
</tr>
<tr>
<td>Dimension_max_depth</td>
<td>int</td>
<td>Profondeur maximale de la dimension.</td>
</tr>
<tr>
<td>Dimension_type</td>
<td>int</td>
<td>Type de la dimension :</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0 = normal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 = system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2 = attribute</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 3 = user info</td>
</tr>
</tbody>
</table>

Propriétés du tPaloDimensionList Standard

Ces propriétés sont utilisées pour configurer le tPaloDimensionList s'exécutant dans le framework de Jobs Standard.
Le composant tPaloDimensionList Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.
## Basic settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste <strong>Component List</strong> présente uniquement les composants de connexion du Job du même niveau.</td>
</tr>
<tr>
<td><strong>Host Name</strong></td>
<td>Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte.</td>
</tr>
<tr>
<td><strong>Server Port</strong></td>
<td>Saisissez le numéro du port d’écoute du serveur Palo.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez le nom de la base de données dans laquelle se trouvent les dimensions qui vous intéressent.</td>
</tr>
<tr>
<td><strong>Retrieve cube dimensions</strong></td>
<td>Cochez cette case pour récupérer les informations des dimensions d’un cube existant.</td>
</tr>
<tr>
<td><strong>Cube</strong></td>
<td>Saisissez le nom du cube duquel les informations concernant les dimensions sont récupérées.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /> <strong>Remarque :</strong></td>
<td>Disponible lorsque vous cochez la case <strong>Retrieve cube dimensions</strong>.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <strong>No</strong> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre <strong>[Repository Content]</strong>.</td>
</tr>
</tbody>
</table>
**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | **DIMENSIONNAME** : nom de la dimension. Cette variable est une variable Flow et retourne une chaîne de caractères.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**. |

**Utilisation**

| Règle d’utilisation | Ce composant peut être utilisé en standalone ou comme composant de début d’un processus. |

| Connections | |

| Limitation | Le schéma de sortie est fixe et en lecture seule.  
Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com). |
Scénario : Récupérer des informations détaillées concernant les dimensions d'une base de données déterminée

Le Job de ce scénario récupère les informations de toutes les dimensions d'une base de données déterminée.

Pour reproduire ce scénario, procédez comme suit :

**Construire le Job**

**Procédure**

1. Déposez un composant `tPaloDimensionList` et un `tLogRow` de la Palette dans l'espace de modélisation graphique.
2. Cliquez-droit sur le `tPaloDimensionList` pour ouvrir le menu contextuel.
3. Dans le menu, sélectionnez `Row > Main` pour relier les deux composants.

**Configurer le composant tPaloDimensionList**

**Procédure**

1. Double-cliquez sur le composant `tPaloDimensionList` pour ouvrir sa vue `Component`.

2. Dans le champ Host name, saisissez le nom de l’hôte ou l’adresse ID du serveur hôte, `localhost` dans cet exemple.
3. Dans le champ Server Port, saisissez le numéro du port d’écoute du serveur Palo. Dans ce scénario, le numéro est 7777.
4. Dans les champs Username (identifiant) et Password (mot de passe), saisissez vos informations d’authentification. Dans cet exemple, les deux sont `admin`. 
5. Dans le champ **Database**, saisissez le nom de la base de données contenant les dimensions qui vous intéressent, **Biker** dans cet exemple.

**Exécuter le Job**

Appuyez sur **F6** pour exécuter le Job.

Les informations concernant les dimensions de la base de donnée **s Biker** sont récupérées et affichées dans la console de la vue **Run**.

Pour plus d’informations concernant le schéma de sortie, consultez Découvrir le schéma de sortie en lecture seule du tPaloDimensionList à la page 3015.
**tPaloInputMulti**

Ce composant récupère les valeurs stockées ou calculées ainsi que les enregistrements à l’extérieur d’un cube.

Le tPaloInputMulti récupère des données (des éléments et des valeurs) d’un cube Palo.

**Propriétés du tPaloInputMulti Standard**

Ces propriétés sont utilisées pour configurer le tPaloInputMulti s’exécutant dans le framework de Jobs Standard.

Le composant tPaloInputMulti Standard appartient à la famille Business Intelligence.

Le composant de ce framework est toujours disponible.

**Basic settings**

| **Use an existing connection** | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host Name</strong></td>
<td>Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte.</td>
</tr>
<tr>
<td><strong>Server Port</strong></td>
<td>Saisissez le numéro du port d’écoute du serveur Palo.</td>
</tr>
</tbody>
</table>
| **Username et Password** | Informations d’authentification de l’utilisateur de Palo.  
Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Database** | Saisissez le nom de la base de données dans laquelle se trouvent les éléments qui vous intéressent. |
| **Cube** | Saisissez le nom du cube dans lequel les éléments de dimension à récupérer sont stockés. |
| **Cube type** | Sélectionnez le type de cube dans la liste déroulante pour le cube sur lequel l’opération donnée doit s’effectuer :  
- **Normal** : Type de cube "normal", type par défaut.  
- **Attribut** : un cube Attribute sera créé avec la création d’un cube "normal".  
- **System** : Type de cube "système". |
**User Info** : un cube **User Info** sera créé/modifié par ce composant.

<table>
<thead>
<tr>
<th>Commit size</th>
<th>Saisissez le nombre de lignes de chaque lot à récupérer.</th>
</tr>
</thead>
</table>

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


**Cube Query**

Remplissez ce tableau pour préciser les données que vous souhaitez récupérer.

- **Column** : les colonnes du schéma sont ajoutées automatiquement à cette colonne, une fois définies dans l'éditeur du schéma. Les colonnes du schéma sont utilisées pour stocker les éléments de dimensions récupérés.

- **Dimensions** : saisissez le nom de chaque dimension du cube duquel vous souhaitez récupérer les éléments de dimensions.

  **Avertissement** :

  L'ordre des dimensions dans cette colonne doit être cohérent avec l'ordre dans le cube stockant ces dimensions.

- **Elements** : saisissez les éléments de dimensions desquels récupérer les données. Si plusieurs éléments
d’une dimension sont requis, séparez-les avec une virgule.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant requiert un composant de sortie. |

**Connections**

<table>
<thead>
<tr>
<th>Liens de sortie (de ce composant à un autre) :</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row : Main</td>
</tr>
<tr>
<td>Liens d’entrée (d’un autre composant à celui-ci) :</td>
</tr>
<tr>
<td>Row : Iterate.</td>
</tr>
<tr>
<td>Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Limitation**

| Selon l’architecture des systèmes OLAP, une seule valeur (texte ou numérique) peut être récupérée du cube. Les colonnes MEASURE et TEXT sont fixes et en lecture seule. Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton Install dans l’onglet Component. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet Modules de |
Scénario : Récupérer des éléments de dimensions dans un cube donné

Le Job de ce scénario récupère plusieurs éléments de dimensions du cube de démo Palo Sales.

Pour reproduire ce scénario, procédez comme suit :

Construire le Job

Procédure

1. Déposez un composant tPaloInputMulti et un tLogRow de la Palette dans l’espace de modélisation graphique.
2. Cliquez-droit sur le composant tPaloInputMulti afin d’ouvrir son menu contextuel.
3. Dans le menu, sélectionnez Row > Main afin de relier le tPaloInputMulti au tLogRow.

Paramétrer la connexion à une base de données

Procédure

1. Double-cliquez sur le tPaloInputMulti afin d’ouvrir sa vue Component.

3. Dans le champ **Server Port**, saisissez le numéro du port d’écoute du serveur Palo. Dans ce scénario, le numéro est 7777.

4. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d’authentification. Dans cet exemple, l’identifiant et le mot de passe sont *admin*.

**Configurer le Cube Query**

**Procédure**

1. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle est stocké le cube.

2. Dans le champ **Cube**, saisissez le nom du cube dans lequel sont stockées les dimensions qui vous intéressent. Dans ce scénario, le cube est celui de démo, *Sales*.

3. Dans le champ **Cube type**, sélectionnez le type du cube à créer. Choisissez **Normal**, ce qui signifie que le cube sera de type “Normal”, le type par défaut.

Dans l'éditeur du schéma, cliquez sur le bouton [+ ] pour ajouter les lignes nécessaires. Dans cet exemple, ajoutez les lignes correspondant à toutes les dimensions stockées dans le cube Sales: Products, Regions, Months, Years, Datatypes, Measures. Saisissez les noms dans l’ordre donné de ce cube.

Cliquez sur OK pour valider les modifications et acceptez la propagation du changement au composant suivant. Ces colonnes sont ajoutées automatiquement dans la colonne Column du tableau Cube Query dans la vue Component. Si l’ordre n’est pas cohérent par rapport à celui du cube Sales, déplacez les colonnes à l’aide des flèches montante et descendante.

Dans la colonne Dimensions du tableau Cube Query, saisissez le nom de chaque dimension stockée dans le cube Sales selon l’ordre des lignes dans la colonne Column. Dans le cube Sales, le nom des dimensions est : Products, Regions, Months, Years, Datatypes, Measures.

Dans la colonne Elements du tableau Cube Query, saisissez le nom des éléments de dimensions que vous souhaitez récupérer selon la dimension à laquelle ils appartiennent. Dans cet exemple, les éléments à récupérer sont All products, Germany, Austria, Jan, 2009, Actual, Turnover (Germany et Austria appartenant à la même dimension Regions, ils doivent être saisis dans la même ligne et séparés par une virgule).

Exécuter le Job

Procédure

1. Cliquez sur le tLogRow afin d’ouvrir sa vue Component.
2. Dans la zone Mode, sélectionnez l’option Table (print values in cell of a table) pour afficher les résultats de l’exécution sous forme de tableau.

3. Appuyez sur F6 pour exécuter le Job.

**Résultats**

Les éléments de dimensions et la valeur Measure correspondant s’affichent dans la console de la vue Run.

```
[statistics] connecting to socket on port 3338
[statistics] connected

+--------------------------------------------------+
<table>
<thead>
<tr>
<th>Products</th>
<th>Regions</th>
<th>Months</th>
<th>Years</th>
<th>Datatypes</th>
<th>Measures</th>
<th>MEASURE</th>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Products</td>
<td>Austria</td>
<td>Jan</td>
<td>2009</td>
<td>Actual</td>
<td>Turnover</td>
<td>4?6977.95</td>
<td>null</td>
</tr>
<tr>
<td>All Products</td>
<td>Germany</td>
<td>Jan</td>
<td>2009</td>
<td>Actual</td>
<td>Turnover</td>
<td>1?334.56</td>
<td>null</td>
</tr>
</tbody>
</table>

[statistics] disconnected
Job retrieve ended at 13:49 02/11/2010. [exit code=0]
```
**tPaloOutput**

Ce composant prend le flux d'entrée et l'écrit dans un Cube Palo donné.

Le tPaloOutput écrit une ligne de données (des éléments ainsi que des valeurs) dans un cube Palo donné.

**Propriétés du tPaloOutput Standard**

Ces propriétés sont utilisées pour configurer le tPaloOutput s'exécutant dans le framework de Jobs Standard.

Le composant tPaloOutput Standard appartient à la famille Business Intelligence.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.</td>
</tr>
<tr>
<td><strong>Host Name</strong></td>
<td>Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte.</td>
</tr>
<tr>
<td><strong>Server Port</strong></td>
<td>Saisissez le numéro du port d'écoute du serveur Palo.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d'authentification de l’utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez le nom de la base de données dans laquelle se trouve le cube qui vous intéresse.</td>
</tr>
<tr>
<td><strong>Cube</strong></td>
<td>Saisissez le nom du cube dans lequel les données entrantes sont écrites.</td>
</tr>
<tr>
<td><strong>Commit size</strong></td>
<td>Saisissez le nombre de lignes de chaque lot devant être écrit dans le cube.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
</tbody>
</table>
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


**Column as Measure** : Sélectionnez la colonne du flux d’entrée contenant les valeurs **Measure** ou **Text**.

**Create element if not exist** : Cochez cette case afin de créer l’élément à traiter s’il n’existe pas.

**Save cube at process end** : Cochez cette case afin de sauvegarder le cube dans lequel vous avez écrit les données à la fin du processus.

**Advanced settings**

- **tStatCatcher Statistics** : Cochez cette case pour collecter les données de log au niveau du composant.

**Global variables**

- **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.
- Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.
- Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace**.
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant requiert un composant d’entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connections</strong></td>
<td>Liens de sortie (de ce composant à un autre) :</td>
</tr>
<tr>
<td></td>
<td><strong>Row</strong> : Iterate.</td>
</tr>
<tr>
<td></td>
<td><strong>Trigger</strong> : Run if.</td>
</tr>
<tr>
<td></td>
<td>Liens d’entrée (d’un autre composant à celui-ci) :</td>
</tr>
<tr>
<td></td>
<td><strong>Row</strong> : Main, Reject.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

| Limitation          | Ce composant ne peut écrire qu’une seule ligne de données dans un cube. |
|---------------------| Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com). |

### Scénario associé

Pour un scénario associé, consultez Scénario 1 : Écrire des données dans un cube donné à la page 3033.
tPaloOutputMulti

Ce composant prend le flux d'entrée et l’écrit dans un cube Palo donné.
Le tPaloOutputMulti écrit des données (des éléments et des valeurs) dans un cube Palo.

Propriétés du tPaloOutputMulti Standard

Ces propriétés sont utilisées pour configurer le tPaloOutputMulti s’exécutant dans le framework de Jobs Standard.
Le composant tPaloOutputMulti Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.

Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.
Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
| Host Name | Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte. |
| Server Port | Saisissez le numéro du port d’écoute du serveur Palo. |
| Username et Password | Informations d’authentification de l’utilisateur de Palo. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Database | Saisissez le nom de la base de données dans laquelle se trouve le cube qui vous intéresse. |
| Cube | Saisissez le nom du cube dans lequel les données entrantes sont écrites. |
| Cube type | Sélectionnez le type de cube dans la liste déroulante pour le cube sur lequel l’opération donnée doit s’effectuer :
- Normal : Type de cube "normal", type par défaut.
- Attribut : un cube Attribut sera créé avec la création d’un cube "normal".
- System : Type de cube "système".
- User Info : un cube User Info sera créé/modifié par ce composant. |
<table>
<thead>
<tr>
<th>Commit size</th>
<th>Saisissez le nombre de lignes de chaque lot devant être écrit dans le cube.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <code>line</code> lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode <code>Repository</code>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <code>Built-In</code> et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <strong>No</strong> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Measure value</strong></td>
<td>Sélectionnez la colonne du flux d’entrée contenant les valeurs <code>Measure</code> ou <code>Text</code>.</td>
</tr>
<tr>
<td><strong>Splash mode</strong></td>
<td>Sélectionnez le mode splash utilisé pour écrire des données dans un élément consolidé. Le mode peut être :</td>
</tr>
<tr>
<td></td>
<td>• <strong>Add</strong> : écrit des valeurs dans les éléments sous-jacents.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default</strong> : utilise le mode splash par défaut.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Set</strong> : définit simplement la valeur et effectue la distribution à partir d’autres valeurs.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Disable</strong> : n’applique aucun splash.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les modes splash Palo, consultez le Guide utilisateur de Palo.</td>
</tr>
<tr>
<td><strong>Add values</strong></td>
<td>Cochez cette case pour ajouter des nouvelles valeurs aux valeurs actuelles. Si vous ne cochez pas cette case, les nouvelles valeurs écraseront les valeurs actuelles.</td>
</tr>
<tr>
<td><strong>Use eventprocessor</strong></td>
<td>Cochez cette case pour appeler le serveur de supervision.</td>
</tr>
</tbody>
</table>
### Die on error

Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

### Global Variables


### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant requiert un composant de début.</th>
</tr>
</thead>
</table>

### Connections

| Liens de sortie (de ce composant à un autre) : | Row : Main  
| Trigger : Run if, On Component Ok, On Component Error. |

| Liens d’entrée (d’un autre composant à celui-ci) : | Row : Main, Reject |

Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.

### Limitation

| Les mesures numériques doivent être de type Double ou String. Lorsque le type String est utilisé, saisissez entre guillemets la valeur devant être traitée. Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton Install dans l’onglet Component. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet Modules de la perspective Integration de votre studio. Vous pouvez |
Scénario 1 : Ecrire des données dans un cube donné

Le Job de ce scénario écrit de nouvelles valeurs dans le cube Sales donné en démo dans la base de données Demo installée avec Palo.

Pour reproduire ce scénario, procédez comme suit :

**Construire le Job**

**Procédure**

1. Déposez un tFixedFlowInput et un tPaloOutputMulti de la Palette dans l'espace de modélisation graphique.
2. Cliquez-droit sur le composant tFixedFlowInput afin d'ouvrir son menu contextuel.
3. Dans ce menu, sélectionnez Row > Main pour relier ce composant au tPaloOutputMulti.

**Configurer le composant d'entrée**

**Procédure**

1. Double-cliquez sur le tFixedFlowInput pour ouvrir sa vue Component.

   ![Configuration de tFixedFlowInput](image)

   **Basic settings**
   - Schema
   - Built-In
   - Edit schema
   - Number of rows: 1
   - Mode: Use Single Table
   - Values:
     - Column: Products, Value: "Desktop L"
     - Region: Regions, Value: "Germany"
     - Months: Months, Value: "Jan"
     - Years: Years, Value: "2009"
     - Datatypes: Datatypes, Value: "Actual"
     - Measures: Measures, Value: "Turnover"
     - Value: 1234.56

   ![Options de configuration](image)

   **Use InLine Table**
   **Use InLine Content delimited file**
2. Cliquez sur le bouton [...] pour ouvrir l'éditeur de schéma.

3. Dans l'éditeur de schéma, cliquez sept fois sur le bouton [+]. Afin d'ajouter sept lignes. Renommez-les respectivement Products, Regions, Months, Years, Datatypes, Measures et Values. L'ordre de ces lignes doit être cohérent avec celui des dimensions correspondantes dans le cube Sales et le type de la colonne Value, qui comprend la valeur de la mesure, est défini comme double/Double.

4. Cliquez sur OK afin de valider la modification et acceptez la propagation proposée par la boîte de dialogue qui s'ouvre. Les libellés des colonnes du schéma s'affichent automatiquement dans le tableau Value sous la case Use single table de la zone Mode.


**Configurer le composant de sortie**

**Procédure**

1. Double-cliquez sur le composant tPaloOutputMulti afin d'ouvrir sa vue Component.
2. Dans le champ **Server Port**, saisissez le numéro du port d'écoute du serveur Palo. Dans ce scénario, le numéro est 7777.

3. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.


5. Dans le champ **Cube**, saisissez le nom que vous souhaitez utiliser pour le cube à créer, *bikerTalend*, par exemple.

6. Dans le champ **Cube type**, sélectionnez le type **Normal** dans la liste déroulante. Cela signifie que le cube à créer sera de type "Normal", le type par défaut.

7. Dans le champ **Measure Value**, sélectionnez *Value*.

**Exécuter le Job**

Appuyez sur *F6* pour exécuter le Job.

Le flux d'entrée de données a été écrit dans le cube *Sales*.

---

**Scénario 2 : Refuser un flux de données d'entrée lorsque les éléments à écrire n'existent pas dans un cube donné**

Le Job de ce scénario essaye d'écrire des données dans le cube *Sales* mais puisque les éléments concernés n'existent pas dans ce cube, le flux d'entrée est refusé.

Pour reproduire ce scénario, procédez comme suit :
Construire le Job

Procédure
1. Déposez les composants `tFixedFlowInput`, `tPaloCheckElements`, `tPaloOutputMulti` et `tLogRow` de la Palette dans l'espace de modélisation graphique.
2. Cliquez-droit sur le `tFixedFlowInput` afin d'ouvrir son menu contextuel.
3. Dans ce menu, sélectionnez `Row > Main` afin de connecter ce composant au `tPaloCheckElements`.

Configurer le composant d'entrée

Procédure
1. Double-cliquez sur le composant `tFixedFlowInput` afin d'ouvrir sa vue `Component`.
2. Cliquez sur le bouton ` [...] ` afin d'ouvrir l'éditeur de schéma.
3. Dans l’éditeur de schéma, cliquez sept fois sur le bouton [+] afin d’ajouter sept lignes que vous allez nommer respectivement *Products*, *Regions*, *Months*, *Years*, *Datatypes*, *Measures* et *Values*. L’ordre de ces lignes doit être cohérent avec celui des dimensions correspondantes dans le cube *Sales* et le type de la colonne *Value*, qui comprend la valeur de la mesure et qui est défini comme double/Double.

4. Cliquez sur **OK** afin de valider la modification et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre. Les libellés des colonnes du schéma s’affichent automatiquement dans le tableau *Value* sous la case *Use single table* de la zone *Mode*.


**Configurer le composant tPaloCheckElements**

**Procédure**

1. Double-cliquez sur le composant **tPaloCheckElements** afin d’ouvrir sa vue *Component*.
2. Dans le champ Host name, saisissez localhost.

3. Dans le champ Server Port, saisissez le numéro du port d’écoute du serveur Palo. Dans ce scénario, le numéro est 7777.

4. Dans les champs Username (identifiant) et Password (mot de passe), saisissez vos informations d’authentification. Dans cet exemple, les deux sont admin.

5. Dans le champ Database, saisissez le nom de la base de données dans laquelle vous souhaitez créer le cube, Demo dans cet exemple.

6. Dans le champ Cube, saisissez le nom du cube dans lequel vous souhaitez écrire les données, Sales, par exemple.

7. Dans le champ On Element Error, sélectionnez Reject dans la liste déroulante.

8. Dans le tableau contenant les éléments en bas de la vue Basic settings, cliquez sur la cellule au croisement de la ligne Value et de la colonne Element type. Sélectionnez Measure dans la liste déroulante.

**Configurer le composant de sortie**

**Procédure**

1. Double-cliquez sur le tPaloOutputMulti pour afficher sa vue Component.
2. Dans le champ **Server Port**, saisissez le numéro du port d’écoute du serveur Palo. Dans ce scénario, le numéro est 7777.

3. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d'authentification. Dans cet exemple, les deux sont *admin*.

4. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle vous souhaitez créer le cube, *Demo* dans cet exemple.


6. Dans le champ **Cube type**, sélectionnez le type **Normal** dans la liste déroulante. Cela signifie que le cube à créer sera de type "Normal", le type par défaut.

7. Dans le champ **Measure Value**, sélectionnez **Value**.

**Exécuter le Job**

Appuyez sur **F6** pour exécuter votre Job.

Les données à écrire sont rejetées et affichées dans la console de la vue Run. Vous pouvez voir que le message d’erreur est *Smart Products*.

```
[statistics] connecting to socket on port 3407
|[--------------------------|---|---|---|---|---|---|---]|
|[Products] [Regions] [Months] [Years] [Datatypes] [Measures] [Value] [errorMessage]
|---------------------------------------------------------------------------|
|[Smart Products] [Germany] [Jan] 2009 [Actual] [Turnover] [1234.56] [Products:Smart Products]
[statistics] disconnected
```

`Job JobOutputExistingElements ended at 14:45 09/11/2010. [exit code=0]`
tPaloRule

Ce composant gère des règles dans un cube Palo donné.
Le tPaloRule crée ou modifie des règles dans un cube Palo donné.

Propriétés du tPaloRule Standard

Ces propriétés sont utilisées pour configurer le tPaloRule s’exécutant dans le framework de Jobs Standard.
Le composant tPaloRule Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.

Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.
Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau. |
| Host Name | Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte. |
| Server Port | Saisissez le numéro du port d’écoute du serveur Palo. |
| Username et Password | Informations d’authentification de l’utilisateur de Palo.
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Database | Saisissez le nom de la base de données dans laquelle se trouvent les dimensions appliquant les règles. |
| Cube | Saisissez le nom du cube dans lequel les informations de dimensions doivent être retrouvées. |
| Cube rules | Remplissez ce tableau afin d’effectuer différentes actions sur les règles spécifiques :
- Definition : saisissez la règle à appliquer. |
- External Id : saisissez l’ID externe personnalisé. |
- Comment : saisissez un commentaire pour cette règle. |
- Activated : cochez cette case afin d’activer la règle. |
- **Action**: sélectionnez dans la liste déroulante l’action à effectuer :
  - **Create**: créer cette règle.
  - **Delete**: supprimer cette règle.
  - **Update**: mettre à jour cette règle.

**Advanced settings**

| TStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | **ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**. |

**Utilisation**

| Règle d’utilisation | Ce composant peut être utilisé en standalone dans la création dynamique de cubes, avec une liste définie de dimensions. |

**Connections**

<table>
<thead>
<tr>
<th>Liens de sortie (de ce composant à un autre) :</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liens d’entrée (d’un autre composant à celui-ci) :</td>
</tr>
<tr>
<td><strong>Row</strong> : Iterate</td>
</tr>
<tr>
<td>Pour plus d’informations concernant les connexions, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

**Limitation**

| La mise à jour et la suppression d’une règle ne sont disponibles uniquement lorsque la règle a été créée avec un ID externe. |
| Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas disponibles. |
Scénario : Créer une règle dans un cube donné

Le Job de ce scénario crée une règle appliquée aux dimensions d’un cube donné.

Pour reproduire ce scénario, procédez comme suit :

**Paramétrer la connexion à une base de données**

**Procédure**

1. Déposez un composant **tPaloRule** de la **Palette** dans l’espace de modélisation graphique.
2. Double-cliquez sur le composant **tPaloRule** afin d’ouvrir sa vue **Component**.

4. Dans le champ **Server Port**, saisissez le numéro du port d’écoute du serveur Palo. Dans ce scénario, le numéro est 7777.

5. Dans les champs **Username** (identifiant) et **Password** (mot de passe), saisissez vos informations d’authentification. Dans cet exemple, les deux sont *admin*.

6. Dans le champ **Database**, saisissez le nom de la base de données dans laquelle se trouvent les dimensions appliquant les règles, *Biker* dans cet exemple.

7. Dans le champ **Cube**, saisissez le nom du cube auquel appartiennent les dimensions appliquant les règles, *Orders* dans cet exemple.

### Définir les règles du cube

**Procédure**

1. Sous le tableau **Cube rules**, cliquez sur le bouton [*] afin d’ajouter une ligne.

2. Dans le tableau **Cube rules**, saisissez [*’2009’*] = 123 dans la colonne **Definition**, *OrderRule1* dans la colonne **External Id** et *Palo Demo Rules* dans la colonne **Comment**.

3. Dans la colonne **Activated**, cochez la case.

4. Dans la colonne **Action**, sélectionnez **Create** dans la liste déroulante.

### Exécuter le Job

Appuyez sur **F6** afin d’exécuter le Job.

La nouvelle règle a été créée et la valeur de chaque élément de 2009 est 123.
tPaloRuleList

Ce composant liste toutes les règles, les formules, les commentaires, les statuts d’activation, les IDs externes d’un cube donné.
Le tPaloRuleList récupère les informations d’une liste de règles d’une base de données Palo spécifiée.

Découvrir le schéma de sortie en lecture seule du tPaloRuleList

Le tableau ci-dessous présente les informations relatives au schéma de sortie en lecture seule du composant tPaloRuleList.

<table>
<thead>
<tr>
<th>Base de données</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rule_identifier</td>
<td>long</td>
<td>Identifiant interne de la règle.</td>
</tr>
<tr>
<td>rule_definition</td>
<td>string</td>
<td>Nom de la dimension.</td>
</tr>
<tr>
<td>Dimension_attribute_cube</td>
<td>string</td>
<td>Formule de la règle. Pour plus d’informations concernant cette règle, consultez le Guide utilisateur Palo.</td>
</tr>
<tr>
<td>ruleExternId</td>
<td>string</td>
<td>ID externe personnalisé.</td>
</tr>
<tr>
<td>ruleComment</td>
<td>string</td>
<td>Commentaire utilisateur de la règle.</td>
</tr>
<tr>
<td>ruleActivated</td>
<td>boolean</td>
<td>Indique si la règle a été activée ou non.</td>
</tr>
</tbody>
</table>

Propriétés du tPaloRuleList Standard

Ces propriétés sont utilisées pour configurer le tPaloRuleList s’exécutant dans le framework de Jobs Standard.
Le composant tPaloRuleList Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Name</td>
<td>Saisissez le nom de l’hôte ou l’adresse IP du serveur hôte.</td>
</tr>
<tr>
<td>Server Port</td>
<td>Saisissez le numéro du port d’écoute du serveur Palo.</td>
</tr>
</tbody>
</table>
**Username et Password**
Informations d’authentification de l’utilisateur de Palo.
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Database**
Saisissez le nom de la base de données dans laquelle se trouve le cube qui vous intéresse.

**Cube**
Saisissez le nom du cube dans lequel les informations de règles doivent être retrouvées.

**Schema et Edit Schema**
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métdonnée du schéma dans la fenêtre **[Repository Content]**.

**Advanced settings**

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**
**NB_RULES** : nombre de règles. Cette variable est une variable **After** et retourne un nombre entier.
**EXTERNAL_RULEID** : ID externe de la règle. Cette variable est une variable *Flow* et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

---

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en standalone ou en tant que composant de début d’un processus.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Connections</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Limitation</th>
<th>Le schéma de sortie est fixe et en lecture seule.</th>
</tr>
</thead>
</table>

---

**Scénario : Récupérer des informations détaillées concernant des règles d’un cube donné**

Le Job de ce scénario récupère les informations concernant les règles appliquées dans les dimensions d’un cube donné.

Pour reproduire ce scénario, procédez comme suit :

- **Scénario : Récupérer des informations détaillées concernant des règles d’un cube donné**

  Le Job de ce scénario récupère les informations concernant les règles appliquées dans les dimensions d’un cube donné.

  Pour reproduire ce scénario, procédez comme suit :
**Construire le Job**

**Procédure**

1. Déposez un composant `tPaloRuleList` et un `tLogRow` de la Palette dans l’espace de modélisation graphique.
2. Cliquez-droit sur le `tPaloRuleList` afin d’ouvrir son menu contextuel.
3. Dans ce menu, sélectionnez `Row > Main` afin de lier les deux composants.

**Configurer le composant `tPaloRuleList`**

**Procédure**

1. Double-cliquez sur le `tPaloRuleList` pour ouvrir sa vue `Component`.

![Configurer le composant tPaloRuleList](image)

2. Dans le champ `Host name`, saisissez le nom de l’hôte ou l’adresse IP du serveur hôte, `localhost` dans cet exemple.
3. Dans le champ `Server Port`, saisissez le numéro du port d’écoute du serveur Palo, `7777` dans cet exemple.
4. Dans les champs `Username` (identifiant) et `Password` (mot de passe), saisissez vos informations d’authentification. Dans cet exemple, les deux sont `admin`.
5. Dans le champ `Database`, saisissez le nom de la base de données dans laquelle se trouvent les dimensions appliquant les règles qui vous intéressent, `Biker` dans cet exemple.
6. Dans le champ `Cube`, saisissez le nom du cube auquel les règles qui vous intéressent appartiennent.

**Exécuter le Job**

Appuyez sur `F6` pour exécuter le Job.

Les informations de toutes les règles du cube `Orders` sont récupérées et affichées dans la console de la vue `Run`. 
Pour plus d’informations concernant le schéma de sortie, consultez Découvrir le schéma de sortie en lecture seule du tPaloRuleList à la page 3044.
**tParAccelBulkExec**

Ce composant améliore les performances pendant les opérations d'INSERT dans une base de données ParAccel.

Les tParAccelOutputBulk et tParAccelBulkExec sont généralement utilisés ensemble pour d'une part générer en sortie le fichier qui sera d'autre part utilisé comme paramètre dans l'exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tParAccelOutputBulkExec, détaillé dans une section séparée. L'intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

Le tParAccelBulkExec effectue une action d'INSERT sur les données fournies.

**Propriétés du tParAccelBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tParAccelBulkExec s'exécutant dans le framework de Jobs Standard.

Le composant tParAccelBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

<i>Remarque : </i>

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

<i>Remarque : </i>

Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

| **Host** | Adresse IP du serveur de base de données. |
| **Port** | Numéro du port d’écoute du serveur de base de données. |
| **Database** | Nom de la base de données. |
| **Schema** | Nom du schéma. |
| **Username et Password** | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Table** | Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit exister pour que l’opération d’Insert soit autorisée. |
| **Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
**None** : n’effectuer aucune opération de table.  
**Drop and create table** : supprimer la table puis en créer une nouvelle.  
**Create table** : créer une table qui n’existe pas encore.  
**Create table if not exists** : créer la table si nécessaire.  
**Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.  
**Clear table** : supprimer le contenu de la table. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.  
**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**. |
Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Advanced settings

| Copy mode | Sélectionnez le mode de Copy que vous souhaitez utiliser.  
| **Basic** : Mode classique, sans optimisation.  
| **Parallel** : Mode permettant d'utiliser plusieurs API internes à ParAccel, afin d'améliorer les performances de chargement. |
| Filename | Nom du fichier à charger.  
| **Avertissement** :  
<p>| Ce fichier est situé sur la machine spécifiée par l'URI dans le champ <strong>Host</strong> et doit être sur la même machine que le serveur de la base de données. |
| File Type | Sélectionnez dans la liste le type de fichier. |
| Field Layout | Sélectionnez dans la liste la disposition du fichier. |
| Field separator | Caractère, chaîne ou expression régulière séparant les champs (si vous chargez un fichier délimité). |</p>
<table>
<thead>
<tr>
<th>Explicit IDs</th>
<th>L’ID sera assigné directement par la base de données ou est déjà présent dans le fichier à charger.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove Quotes</td>
<td>Cochez cette case afin de retirer les guillemets de votre fichier à charger.</td>
</tr>
<tr>
<td>Max. Errors</td>
<td>Saisissez le nombre maximal d’erreurs avant que votre Job ne s’arrête.</td>
</tr>
<tr>
<td>Date Format</td>
<td>Saisissez le format de date qui sera utilisé.</td>
</tr>
<tr>
<td>Time/Timestamp Format</td>
<td>Saisissez le format de date et heure qui sera utilisé.</td>
</tr>
<tr>
<td>Additional COPY Options</td>
<td>Saisissez l’option spécifique à ParAccel et personnalisée que vous souhaitez utiliser.</td>
</tr>
<tr>
<td>Log file</td>
<td>Saisissez le chemin d’accès à votre fichier de log, ou parcourrez votre répertoire.</td>
</tr>
<tr>
<td>Logging level</td>
<td>Sélectionnez le niveau de verbose à retourner.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé avec un composant tParAccelOutputBulk. Ensemble, ils offrent un gain de performance important pour l’alimentation d’une base de données ParAccel.

**Dynamic settings**

Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les...
<table>
<thead>
<tr>
<th>paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Limitation</strong></td>
</tr>
</tbody>
</table>
| Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

### Scénarios associés

Pour plus d’informations relatives au fonctionnement de **tParAccelBulkExec**, consultez les scénarios suivants dans :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du composant **tMysqlOutputBulk**.
- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du composant **tMysqlOutputBulkExec**.
- **Scénario : Supprimer et insérer des données dans une base Oracle** à la page 2914 du composant **tOracleBulkExec**.
tParAccelClose

Ce composant ferme une connexion à la base de données ParAccel.
Le tParAccelClose ferme la connexion à une base de données connectée.

Propriétés du tParAccelClose Standard

Ces propriétés sont utilisées pour configurer le tParAccelClose s’exécutant dans le framework de Jobs Standard.
Le composant tParAccelClose Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tParAccelConnection dans la liste.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple... |
ple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tParAccelCommit

Commite en une seule fois une transaction globale, en utilisant une connexion unique, au lieu de commiter chaque ligne ou chaque lot de lignes et améliore ainsi les performances.
Le tParAccelCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tParAccelCommit Standard

Ces propriétés sont utilisées pour configurer le tParAccelCommit s’exécutant dans le framework de Jobs Standard.
Le composant tParAccelCommit Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tParAccelConnection dans la liste.</td>
</tr>
</tbody>
</table>
| Close connection | Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. Avertissement :
Si vous utilisez un lien de type Row >Main pour relier le tParAccelCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants ParAccel et notamment <strong>tParAccelConnection</strong> et <strong>tParAccelRollback</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez <strong>Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte</strong> à la page 2641 et <strong>Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement</strong> à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé au composant **tParAccelCommit**, consultez **Scénario : Insérer des données dans des tables mère/fille** à la page 2620.
tParAccelConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Le tParAccelConnection ouvre une connexion vers une base de données afin d’effectuer une transaction.

Propriétés du tParAccelConnection Standard

Ces propriétés sont utilisées pour configurer le tParAccelConnection s’exécutant dans le framework de Jobs Standard.

Le composant tParAccelConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td></td>
<td><em>Built-in</em> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><em>Repository</em> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
### Use or register a shared DB Connection

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un composant **tRunJob** ayant une de ces options activée fera échouer votre Job.

### Advanced settings

#### Auto Commit

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commit pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant **Commit**.

#### tStatCatcher Statistics

Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Utilisation

#### Règle d’utilisation

Ce composant est généralement utilisé avec des composants **ParAccel**, notamment les composants **tParAccelCommit** et **tParAccelRollback**.

#### Limitation

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio.
Scénario associé

Pour un scénario associé au composant **tParAccelConnection**, consultez **tMysqlConnection** à la page 2618.
tParAccelInput

Ce composant lit une base de données et en extrait des champs à l'aide de requêtes.

Le tParAccelInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

Propriétés du tParAccelInput Standard

Ces propriétés sont utilisées pour configurer le tParAccelInput s'exécutant dans le framework de Jobs Standard.

Le composant tParAccelInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

| Database | Sélectionnez un type de base de données dans la liste et cliquez sur Apply. |
| Property type | Peut être Built-in ou Repository |
| **Built-in** | Propriétés utilisées ponctuellement. |
| **Repository** | Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |
| | Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. |
| | Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend. |
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |
| **Remarque** | Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une |
connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Nom exact du schéma</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs. **Built-in** : Le schéma sera créé et conservé pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**. **Repository** : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
  • **View schema** : sélectionnez cette option afin de voir le schéma.
  • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
  • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans... |
le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th>Table name</th>
<th>Nom de la table de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query type et Query</strong></td>
<td>Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs correspond à celui défini dans le schéma.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Use cursor</th>
<th>Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d’optimiser les performances.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trim all the String/Char columns</strong></td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td><strong>Trim column</strong></td>
<td>Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| Global Variables | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable *After* et retourne un entier.  
**QUERY** : requête traitée. Cette variable est une variable *Flow* et retourne une chaîne de caractères.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------|
## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données ParAccel.</th>
</tr>
</thead>
</table>
| **Dynamic settings** | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.  


## Scénarios associés

Pour un scénario associé, consultez : 
tParAccelOutput

Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.
Le tParAccelOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

**Propriétés du tParAccelOutput Standard**

Ces propriétés sont utilisées pour configurer le tParAccelOutput s’exécutant dans le framework de Jobs Standard.

Le composant tParAccelOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><img src="image" alt="Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant." /></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><img src="image" alt="Remarque :" /></td>
<td>Remarque : Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par</td>
</tr>
</tbody>
</table>
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur de la base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Table      | Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois. |
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
**None** : n’effectuer aucune opération de table.  
**Drop and create the table** : supprimer la table puis en créer une nouvelle.  
**Create a table** : créer une table qui n’existe pas encore.  
**Create table if doesn’t exist** : créer la table si nécessaire.  
**Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.  
**Clear a table** : supprimer le contenu de la table. |
| Action on data | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :  
**Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.  
**Update** : Mettre à jour les entrées existantes.  
**Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. |
### Update or insert

Met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

### Delete

Supprime les entrées correspondantes au flux d’entrée.

#### Avertissement :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations Update et Delete. Pour cela, cliquez sur le bouton [...] à côté du champ Edit Schema et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet Advanced settings pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case Use field options et sélectionnez la case Key in update correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases Key in delete pour les opérations de suppression (Delete).

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

#### Built-In

Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

#### Repository

Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Cliquez sur Edit schema pour modifier le schéma.

Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

| Die on error | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**. |

### Advanced settings

| Commit every | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution. |
| Additional Columns | Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez précédemment supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier. |

| Name | Saisissez le nom de la colonne à modifier ou à insérer. |
| SQL expression | Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes. |
| Position | Sélectionnez **Before, Replace** ou **After**, en fonction de l’action à effectuer sur la colonne de référence. |
| Reference column | Saisissez une colonne de référence que le composant **tParAccelOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier. |

| Use field options | Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données. |
| Use Batch | Cochez cette case pour activer le mode de traitement par lots pour le traitement des données. |

**Remarque :**

Cette case est disponible lorsque vous sélectionnez **Insert, Update, ou Delete** dans la liste **Action on data**.
### Batch Size
Spécifiez le nombre d’enregistrements à traiter dans chaque lot.
Ce champ est disponible uniquement lorsque la case **Use batch mode** est cochée.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

---

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable <em>After</em> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

---

### Utilisation

**Règle d’utilisation**
Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données ParAccel. Il permet aussi de créer un flux de rejet avec un lien **Row > Reject** filtrant les données en erreur. Pour un exemple d’utilisation, consultez *Scénario : Récupérer les données erronées à l’aide d’un lien Reject* à la page 2675 du composant *tMysqlOutput*.

**Dynamic settings**
Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de
contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Limitation


Scénarios associés

Pour un scénario associé au composant tParAccelOutput consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667 du tMysqlOutput.
tParAccelOutputBulk

Ce composant prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données ParAccel.

Les tParAccelOutputBulk et tParAccelBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tParAccelOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Le tParAccelOutputBulk écrit un fichier composé de colonnes et basé sur le séparateur défini et sur les standards ParAccel.

Propriétés du tParAccelOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tParAccelOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tParAccelOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom du fichier à générer.</td>
</tr>
<tr>
<td>Append</td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
</tbody>
</table>

Avertissement :

Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.
### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **Repository Content**.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Row separator</strong></th>
<th>Chaîne (ex : &quot;\n&quot; sous Unix) séparant les lignes.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field separator</strong></td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td><strong>Include header</strong></td>
<td>Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.</td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez <strong>Custom</strong> et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</td>
</tr>
</tbody>
</table>
**Global Variables**


| **Utilisation** | Ce composant est généralement utilisé avec le composant tParAccelBulkExec. Ensemble ils offrent un gain de performance important pour l’alimentation d’une base de données ParAccel. |

**Scénarios associés**

Pour un scénario associé au composant tParAccelOutputBulk, consultez :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du tMysqlOutputBulk.
- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec.
- Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du tOracleBulkExec.
tParAccelOutputBulkExec

Ce composant améliore les performances pendant les opérations d'Insert dans une base de données ParAccel.

Les tParAccelOutputBulk et tParAccelBulkExec sont généralement utilisés ensemble comme deux parties d'un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l'opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tParAccelOutputBulkExec.

Le tParAccelOutputBulkExec effectue une action d'Insert sur les données fournies.

Propriétés du tParAccelOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tParAccelOutputBulkExec s'exécutant dans le framework de Jobs Standard.

Le composant tParAccelOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de la base de données.</td>
</tr>
<tr>
<td>Seuls localhost, 127.0.0.1 ou l'adresse IP exacte de la machine locale permettent un fonctionnement optimal. Le serveur de la base de données doit être installé sur la même machine que le Studio Talend ou que le Job contenant un tParAccelOutputBulkExec.</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma.</td>
</tr>
</tbody>
</table>
### Username et Password
Informations d'authentification de l'utilisateur de base de données.
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

### Table
Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit déjà exister pour que l’opération d’Insert soit autorisée.

### Action on table
Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :
- **None**: n’effectuer aucune opération de table.
- **Drop and create table**: supprimer la table puis en créer une nouvelle.
- **Create table**: créer une table qui n’existe pas encore.
- **Create table if not exists**: créer la table si nécessaire.
- **Drop table if exists and create**: supprimer la table si elle existe déjà, puis en créer une nouvelle.
- **Clear table**: supprimer le contenu de la table.

### Schema et Edit Schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

- **Built-In**: Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
- **Repository**: Le schéma existe déjà et il est stocké dans le *Repository*. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center ([https://help.talend.com](https://help.talend.com)).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Advanced settings**

| Copy mode | Sélectionnez le mode de Copy que vous souhaitez utiliser.  
| Basic : Mode classique, sans optimisation.  
<p>| Parallel : Mode permettant d'utiliser plusieurs API internes à ParAccel, afin d'améliorer les performances de chargement. |
| --- | --- |
| Filename | Nom du fichier à traiter et chemin d'accès. |
| File Type | Sélectionnez dans la liste le type de fichier. |
| Field Layout | Sélectionnez dans la liste la disposition du fichier. |
| Field separator | Caractère, chaîne ou expression régulière séparant les champs (si vous chargez un fichier délimité). |
| Explicit IDs | L’ID sera assigné directement par la base de données ou est déjà présent dans le fichier à charger. |
| Remove Quotes | Cochez cette case afin de retirer les guillemets de votre fichier à charger. |
| Max. Errors | Saisissez le nombre maximal d’erreurs avant que votre Job ne s’arrête. |
| Date Format | Saisissez le format de date qui sera utilisé. |
| Time/Timestamp Format | Saisissez le format de date et heure qui sera utilisé. |
| Additional COPY Options | Saisissez l’option spécifique à ParAccel et personnalisée que vous souhaitez utiliser. |
| Log file | Saisissez le chemin d’accès à votre fichier de log, ou parcouruez votre répertoire. |
| Logging level | Sélectionnez le niveau de verbose à retourner. |</p>
<table>
<thead>
<tr>
<th><strong>tStatCatcher Statistics</strong></th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th><strong>Règle d’utilisation</strong></th>
<th>Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Limitation</strong></td>
<td>Le serveur de la base de données doit être installé sur la même machine que le Studio Talend ou que le Job contenant un <strong>tParAccelOutputBulkExec</strong>, afin que le composant fonctionne correctement.</td>
</tr>
</tbody>
</table>

**Scénarios associés**

Pour un scénario associé au composant **tParAccelOutputBulkExec**, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du **tMysqlOutputBulk**.
- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du **tMysqlOutputBulkExec**.
- **Scénario : Supprimer et insérer des données dans une base Oracle** à la page 2914 du **tOracleBulkExec**.
**tParAccelRollback**

Ce composant évite le commit de transaction involontaire.
Le tParAccelRollback annule la transaction dans une base de données connectée.

### Propriétés du tParAccelRollback Standard

Ces propriétés sont utilisées pour configurer le tParAccelRollback s’exécutant dans le framework de Jobs Standard.
Le composant tParAccelRollback Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

<i>Remarque :</i>
Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tParAccelConnection dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Utilisation

**Règle d’utilisation**
Ce composant est généralement utilisé avec d’autres composants ParAccel, notamment les composants tParAccelConnection et tParAccelCommit.

**Dynamic settings**
Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez
Lorsqu'un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

Pour un scénario associé au composant **tParAccelRollback**, consultez **Scénario : Annuler l’insertion de données dans des tables mère/fille** à la page 2623.
tParAccelRow

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler), selon la nature de la requête et de la base de données. Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Le tParAccelRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tParAccelRow Standard

Ces propriétés sont utilisées pour configurer le tParAccelRow s'exécutant dans le framework de Jobs Standard.

Le composant tParAccelRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
la vue **Basic settings** du composant de connexion créant cette connexion.

2. **Au niveau enfant,** utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend.*

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password,</strong> puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <em>line</em> lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong>: Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <em>Guide utilisateur du Studio Talend.</em></td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong>: Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le <em>Guide utilisateur du Studio Talend.</em></td>
</tr>
<tr>
<td></td>
<td>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository,</strong> trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
</tbody>
</table>
|              | • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la
<table>
<thead>
<tr>
<th>Table name</th>
<th>Nom de la table de base à lire.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query type</td>
<td>La requête peut être Built-in ou distante dans le Repository</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.</td>
</tr>
<tr>
<td>Query</td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décrochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Rejects.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Propagate QUERY’s recordset</th>
<th>Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque</strong> :</td>
</tr>
<tr>
<td></td>
<td>Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type Object. Ce composant est généralement suivi du tParseRecordSet.</td>
</tr>
<tr>
<td></td>
<td><strong>Parameter Index</strong> : Saisissez la position du paramètre dans l’instruction SQL.</td>
</tr>
<tr>
<td></td>
<td><strong>Parameter Type</strong> : Saisissez le type du paramètre.</td>
</tr>
<tr>
<td></td>
<td><strong>Parameter Value</strong> : Saisissez la valeur du paramètre.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque</strong> :</td>
</tr>
<tr>
<td></td>
<td>Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.</td>
</tr>
<tr>
<td>Commit every</td>
<td>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

#### Global Variables

**QUERY** : requête traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

#### Règle d’utilisation

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.

#### Dynamic settings

Cliquez sur le bouton [*] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions.
### dynamiques basées sur les variables de contexte

à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

### Limitation


### Scénarios associés

Pour un scénario associé, consultez :

- Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
- Procédure du composant tDBSQLRow.
- Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du composant tMysqlRow.
tParAccelSCD

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée.
Le tParAccelSCD reflète et traque les modifications d’une table ParAccel SCD dédiée.

**Propriétés du tParAccelSCD Standard**

Ces propriétés sont utilisées pour configurer le tParAccelSCD s’exécutant dans le framework de Jobs Standard.
Le composant tParAccelSCD Standard appartient aux familles Business Intelligence et Databases.
Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong> Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
</tr>
</tbody>
</table>
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Sélectionnez dans la liste le pilote à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td>Built-in</td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

### SCD Editor

L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension.

Pour plus d’informations, consultez *Méthodologie de gestion du SCD* à la page 2716.

### Use memory saving Mode

Cochez cette case pour améliorer les performances du système.

### Source keys include Null

Cochez cette case pour autoriser, dans les colonnes clés source, les valeurs *Null*. **Avertissement :**

Lorsque cette case est cochée, assurez-vous que la valeur de(s) clé(s) source est unique.

### Die on error

Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

### Advanced settings

**End date time details**

Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format *HH:mm:ss*. La valeur par défaut pour ce champ est 12:00:00.

Ce champ apparaît uniquement lorsqu’un SCD de *Type 2* est utilisé et lorsque *Fixed year value* est sélectionné pour créer la date de fin du SCD.

**Debug mode**

Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Variables globales

**Global Variables**

- **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable *After* et retourne un entier.
- **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable *After* et retourne un entier.
- **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable *After* et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez <em>Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte</em> à la page 2641 et <em>Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement</em> à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

### Limitation

Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD.

### Scénario associé

Pour un scénario associé, consultez tMysqlSCD à la page 2712.
tParseRecordSet

Ce composant analyse un ensemble d’enregistrements d’une table au lieu de les analyser individuellement.

Le tParseRecordSet se trouve à la racine de la famille Databases de la Palette de la perspective Integration du Studio Talend. Il couvre des besoins indirectement liés à l’utilisation des bases de données de tout type.

Le tParseRecordSet analyse un ensemble d’enregistrements d’une table de données ou d’une requête SQL et retourne éventuellement des enregistrements seuls.

**Propriétés du tParseRecordSet Standard**

Ces propriétés sont utilisées pour configurer le tParseRecordSet s’exécutant dans le framework de Jobs Standard.

Le composant tParseRecordSet Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Prev. Comp. Column list</th>
<th>Sélectionnez la colonne contenant l’ensemble d’enregistrements à analyser.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Built-in</th>
<th>Le schéma est créé et conservé ponctuellement pour ce composant uniquement. Voir également le Guide utilisateur du Studio Talend.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Repository</th>
<th>Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il est réutilisable. Voir également le Guide utilisateur du Studio Talend.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</th>
<th></th>
</tr>
</thead>
</table>

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la
### tParseRecordSet

métadonnée du schéma dans la fenêtre [Repository Content].

| Attribute table | Paramétrez la valeur positionnelle de chaque colonne de chaque enregistrement d’un ensemble d’enregistrements. |

### Advanced settings

| tStatCatcher Statistics | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

### Global Variables


### Utilisation

| Règle d’utilisation | Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d’entrée sont autorisés. |

| Limitation | Ce composant est principalement utilisé avec la fonctionnalité Recordset d’un composant SP. |

### Scénario associé

Pour un exemple d’utilisation du tParseRecordSet, consultez Scénario 2 : Utiliser l’instance PreparedStatement pour faire une requête sur des données à la page 2702.
**tPigAggregate**

Ce composant ajoute une ou plusieurs colonne(s) supplémentaire(s) dans la sortie des données regroupées afin de créer des données à utiliser par Pig.

Le tPigAggregate regroupe les données originales par colonnes et ajoute une ou plusieurs colonne(s) supplémentaire(s) dans la sortie des données regroupées précédentes.

**Propriétés du tPigAggregate Standard**

Ces propriétés sont utilisées pour configurer le tPigAggregate s’exécutant dans le framework de Jobs Standard.

Le composant tPigAggregate Standard appartient aux familles Big Data et Processing.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.  
Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :
  - **View schema** : sélectionnez cette option afin de voir le schéma.
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre *Repository Content*.

| Built-in | Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
### Group by
Cliquez sur le bouton [+ ] pour ajouter une ou plusieurs colonne(s) afin de définir les tuples dans les données source comme conditions de groupe.

### Operations
Cliquez sur le bouton [+ ] pour ajouter une ou plusieurs colonne(s) afin de générer une ou plusieurs colonne(s) de sortie à partir de conditions :
- **Additional Output Column** : Choisissez une colonne dans les données originales comme colonne de sortie.
- **Function** : Les fonctions des opérations sur les données d'entrée.
- **Input Column** : Choisissez une colonne dans les données originales comme colonne d'entrée.

### Advanced settings
- **Increase parallelism**
  Cochez cette case pour définir le nombre de tâches "reduce" pour les Jobs MapReduce.
- **tStatCatcher Statistics**
  Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu'au niveau de chaque composant.

### Global Variables
- **ERROR MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.
  Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
  Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
  Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation
- **Règle d’utilisation**
  Ce composant est généralement utilisé comme étape intermédiaire. Il nécessite donc un composant d’entrée et un composant de sortie.
- **Prérequis**
  La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.
• Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

• Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

| Limitation | La connaissance des scripts Pig est nécessaire. |

Scénario associé

Pour plus d’informations concernant le fonctionnement du composant tPigAggregate, consultez Agréger des valeurs et trier des données à la page 125.
tPigCode

Ce composant étend les fonctionnalités d'un Job **Talend** via des scripts Pig.

Le tPigCode saisit du code Pig personnalisé afin de l'intégrer dans un programme **Talend**. Vous pouvez exécuter ce code une seule fois.

**Propriétés du tPigCode Standard**

Ces propriétés sont utilisées pour configurer le tPigCode s'exécutant dans le framework de Jobs Standard.

Le composant tPigCode Standard appartient aux familles Big Data et Processing.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de **Talend**.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs. Cliquez sur `Edit schema` pour modifier le schéma. Si le schéma est en mode `Repository`, trois options sont disponibles :
|                      |   • **View schema** : sélectionnez cette option afin de voir le schéma. 
|                      |   • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. 
|                      |   • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. |

| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*. |
| Repository | Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. |

| Scripts | Saisissez les scripts Pig que vous souhaitez exécuter selon la tâche que vous souhaitez effectuer. Pour plus d'informations concernant la syntaxe des fonctions Pig, consultez la documentation Apache relative aux UDF de Pig (fonctions personnalisées) : |
Les composants Pig écrivent en sortie des tuples et configurent automatiquement un alias pour chaque tuple. Lorsque vous utilisez un tuple dans votre script Pig, vous devez saisir le bon alias.

La syntaxe de l'alias est `Component_ID_row_ID_Result`, par exemple, `tPigCode_1_row2_Result`.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>tStatCatcher Statistics</strong></th>
<th>Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable escape</strong></td>
<td>Cochez cette case pour pouvoir écrire du code Pig brut dans le champ <strong>Scripts</strong> sans devoir garder à l’esprit les caractères d’échappement, habituellement requis pour générer du code Java.</td>
</tr>
</tbody>
</table>

### Global Variables

| **Global Variables** | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**. |

### Utilisation

| **Règle d’utilisation** | Ce composant est généralement utilisé comme étape intermédiaire. Il nécessite donc un composant d’entrée et un composant de sortie.
Un composant *tPigCode* peut seulement exécuter une instruction Pig Latin. Si vous devez en exécuter plusieurs, utilisez le nombre correspondant de composants *tPigCode* et exécutez-les l’un après l’autre.
Si un fichier .jar particulier est requis pour exécuter une instruction, vous devez enregistrer ce fichier de bibliothèque via le composant *tPigLoad* démarrant le processus Pig en question. |

<table>
<thead>
<tr>
<th>Prérequis</th>
</tr>
</thead>
<tbody>
<tr>
<td>La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le <em>Studio Talend</em>. La liste suivante présente des informations d’exemple relatives à MapR.</td>
</tr>
<tr>
<td>• Assurez-vous d’avoir installé le client MapR sur la même machine que le <em>Studio Talend</em> et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est <em>lib\MapRClient.dll</em> dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : <a href="http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr">http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr</a> (en anglais).</td>
</tr>
<tr>
<td>Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : <em>no MapRClient in java.library.path.</em></td>
</tr>
<tr>
<td>• Configurez l’argument <code>-Djava.library.path</code>, par exemple, dans la zone <em>Job Run VM arguments</em> de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (<em>Data viewer</em>) afin de visualiser localement dans le studio les données stockées dans MapR.</td>
</tr>
<tr>
<td>Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>La connaissance des scripts Pig est nécessaire.</td>
</tr>
</tbody>
</table>

**Scénario : Sélectionner une colonne de données d’un fichier d’entrée et la stocker dans un fichier local**

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario décrit un Job à trois composants sélectionnant une colonne de données correspondant à une condition de filtre définie dans le **tPigCode** et stocke le résultat dans un fichier local.
**Construire le Job**

**Procédure**

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : tPigCode, tPigLoad et tPigStoreResult.
2. Cliquez-droit sur le tPigLoad afin de le relier au composant tPigCode à l'aide d'un lien Row > Pig Combine.
3. Cliquez-droit sur le tPigCode pour le connecter au tPigStoreResult à l'aide d'un lien Row > Pig Combine.

**Charger les données**

**Procédure**

1. Double-cliquez sur le composant tPigLoad afin d'ouvrir sa vue Basic settings.

   ![Image of tPigLoad settings](image)

   2. Cliquez sur le bouton [..] à côté du champ Edit schema pour ajouter des colonnes au tPigLoad.

   ![Image of adding columns](image)

   3. Cliquez sur le bouton [+ ] afin d’ajouter les colonnes Name, Country et Age puis cliquez sur OK afin de sauvegarder la configuration.

4. Sélectionnez Local dans la zone Mode.
5. Dans le champ Input file URI, renseignez le chemin d'accès complet au fichier d'entrée.
Dans ce scénario, le fichier d’entrée est *CustomerList* et contient des lignes de noms, des noms de pays et des âges.

6. Sélectionnez *PigStorage* dans la liste *Load function*.
7. Laissez les autres paramètres tels qu’ils sont.

**Configurer le composant tPigCode**

**Procédure**

1. Double-cliquez sur le composant *tPigCode* afin d’ouvrir sa vue *Basic settings*.

   ![Diagram](image.png)

2. Cliquez sur *Sync columns* pour récupérer la structure du schéma du composant précédent.
3. Dans le champ *Script Code* saisissez l’expression suivante :

   ```
   tPigCode_1_row2_RESULT = foreach tPigLoad_1_row1RESULT generate $0 as name;
   ```

   Cette expression de filtre sélectionne la colonne *Name* dans *CustomerList*.

**Sauvegarder les données de résultat dans un fichier local**

**Procédure**

1. Double-cliquez sur le composant *tPigStoreResult* afin d’ouvrir sa vue *Basic settings*.

2. Cliquez sur *Sync columns* afin de récupérer la structure du schéma du composant précédent.
3. Dans le champ *Result file*, saisissez le chemin d’accès au fichier de résultats, ou parcourez votre système jusqu’à ce fichier.

   Dans ce scénario, le résultat est sauvegardé dans le fichier *Result*. 

---

3098
4. Cochez la case **Remove result directory if exists**.
5. Sélectionnez **PigStorage** dans la liste **Store function**.
6. Laissez les autres paramètres tels qu’ils sont.

**Exécuter le Job**

Sauvegardez votre Job et appuyez sur **F6** pour l’exécuter.

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
</tr>
<tr>
<td>Silvia</td>
</tr>
<tr>
<td>RONEO</td>
</tr>
<tr>
<td>Ahmad</td>
</tr>
<tr>
<td>Toyota</td>
</tr>
<tr>
<td>Manik</td>
</tr>
<tr>
<td>Natasha</td>
</tr>
<tr>
<td>Billy</td>
</tr>
<tr>
<td>Eminem</td>
</tr>
<tr>
<td>Bill</td>
</tr>
<tr>
<td>Fernicka</td>
</tr>
<tr>
<td>Huamei</td>
</tr>
<tr>
<td>Selena</td>
</tr>
<tr>
<td>Julio</td>
</tr>
<tr>
<td>Pantalion</td>
</tr>
<tr>
<td>Simao</td>
</tr>
<tr>
<td>Nancy</td>
</tr>
<tr>
<td>Gaddafi</td>
</tr>
<tr>
<td>Zidane</td>
</tr>
<tr>
<td>Didi</td>
</tr>
<tr>
<td>Juan</td>
</tr>
<tr>
<td>Bob</td>
</tr>
<tr>
<td>Mario</td>
</tr>
<tr>
<td>Ricky</td>
</tr>
</tbody>
</table>

Le fichier **Result** est généré et il contient la colonne de données sélectionnée.
**tPigCoGroup**

Ce composant effectue une opération Pig COGROUP afin de grouper et agréger les données provenant de multiples flux Pig.

Le tPigCoGroup groupe des données d’autant de sources que nécessaire provenant des composants Pig précédents. Il agrège les données groupées à l’aide d’une fonction donnée avant de les envoyer au composant Pig suivant.

**Propriétés du tPigCoGroup Standard**

Ces propriétés sont utilisées pour configurer le tPigCoGroup s’exécutant dans le framework de Jobs Standard.

Le composant tPigCoGroup Standard appartient aux familles Big Data et Processing.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode `Repository`, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode `Built-In` et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


| **Basic settings** | **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode `Repository`, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode `Built-In` et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Group by

Cliquez sur le bouton [+] pour ajouter une ou plusieurs colonne(s) des flux d’entrée à la table Group by, afin de configurer des colonnes comme condition de groupe.

### Output mapping

Cette table est automatiquement renseignée par le schéma de sortie défini dans le champ Schema. Renseignez cette table pour configurer comment les données groupées doivent être agrégées dans le flux de sortie :

- **Function** : sélectionnez la fonction à utiliser pour agréger une colonne donnée.
- **Source schema** : sélectionnez le flux d’entrée duquel agréger les données.
- **Expression** : sélectionnez la colonne à agréable et, si nécessaire, modifiez les expressions.

### Advanced settings

**Group optimization**

 Sélectionnez l’algorithme Pig selon la situation des données d’entrée et le moyen de chargement utilisé pour optimiser l’opération COGROUP.

Pour plus d’informations, consultez la documentation Apache pour Pig.

**Use partitioner**

Cochez cette case pour appeler un partitionneur Hadoop pour partitionner des enregistrements et retourner la ou les tâche(s) Reduce où chaque enregistrement doit aller.

Notez que la classe du partitionneur doit être enregistrée dans la table **Register jar** du composant **tPigLoad** démarrant le processus Pig en cours.

**Increase parallelism**

Cochez cette case pour configurer le nombre de tâches Reduce pour les Jobs **MapReduce**.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant intermédiaire et nécessite un composant d’entrée et un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prérequis</td>
<td>La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.</td>
</tr>
<tr>
<td></td>
<td>• Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les bibliothèques du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : <a href="http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr">http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr</a> (en anglais). Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante: no MapRClient in java.library.path.</td>
</tr>
<tr>
<td></td>
<td>• Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR. Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.</td>
</tr>
</tbody>
</table>

| Limitation           | La connaissance des scripts Pig est nécessaire. |

### Scénario : Agréger des données de deux relations à l’aide d’une opération COGROUP

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Dans ce scénario, un Job comprenant quatre composants est créé pour agréger deux relations sur un cluster Hadoop donné.
Les deux relations utilisées dans ce scénario se présentent comme suit :

1. Alice, turtle, 17
   Alice, goldfish, 17
   Alice, cat, 17
   Bob, dog, 18
   Bob, cat, 18
   John, dog, 19
   Mary, goldfish, 16
   Bill, dog, 20

Cette relation se compose de trois colonnes nommées owner (maître), pet (animal) et age (des maîtres).

2. Cindy, Alice
   Mark, Alice
   Paul, Bob
   Paul, Jane
   John, Mary
   William, Bill

Cette relation fournit une liste de noms d’étudiants ainsi que de leurs amis. Pour certains, la première relation est celle de détenants d’animaux. Le schéma de cette relation contient deux colonnes : student (étudiant) et friend (ami).

Avant de reproduire ce scénario, vous devez écrire les données d’exemple dans le système HDFS du cluster Hadoop à utiliser. Pour ce faire, vous pouvez utiliser un tHDFSOutput. Pour plus d’informations concernant ce composant, consultez tHDFSOutput à la page 1620.

Les données utilisées dans ce scénario sont inspirées des exemples utilisés dans la documentation Pig pour expliquer les opérateurs GROUP et GOGROUP. Pour plus d’informations, consultez la documentation Apache pour Pig.

**Relier les composants**

**Procédure**

1. Dans la perspective Integration du Studio, créez un nouveau Job depuis le nœud Job Designs de la vue Repository.
   Pour plus d’informations sur la création de Jobs, consultez le Guide utilisateur du Studio Talend.
2. Dans l’emplacement graphique, saisissez le nom du composant à utiliser et sélectionnez ce composant dans la liste qui apparaît. Dans ce scénario, les composants utilisés sont :
deux \texttt{tPigLoad}, un \texttt{tPigCoGroup} et un \texttt{tPigStoreResult}. L'un des \texttt{tPigLoad} est utilisé en tant que composant de chargement principal pour vous connecter au cluster Hadoop à utiliser.

3. Reliez le composant \texttt{tPigLoad} au \texttt{tPigCoGroup} à l'aide d'un lien \texttt{Row > Main}.
4. Répétez l'opération pour connecter le second \texttt{tPigLoad} au \texttt{tPigCoGroup}. Le libellé \texttt{Lookup} s'affiche sur ce lien.
5. Répétez l'opération pour connecter le \texttt{tPigCoGroup} au \texttt{tPigStoreResult}.

\textbf{Lire les données dans le flux Pig}

\textbf{Lire les données d’exemples relatives aux détenteurs d’animaux}

\textbf{Procédure}

1. Double-cliquez sur le composant \texttt{tPigLoad} principal pour ouvrir sa vue \texttt{Component}.
2. Cliquez sur le bouton […] à côté du champ \texttt{Edit schema} pour ouvrir l’éditeur du schéma puis cliquez trois fois sur le bouton [+] pour ajouter trois lignes.
3. Dans la colonne \texttt{Column}, renommez les nouvelles lignes \textit{owner}, \textit{pet} et \textit{age}, respectivement. Dans la colonne \texttt{Type} de la ligne \textit{age}, sélectionnez \texttt{Integer}.
4. Cliquez sur \texttt{OK} pour valider les modifications et accepter la propagation proposée par la boîte de dialogue qui s’ouvre.
5. Dans la zone \texttt{Mode}, sélectionnez \texttt{Map/Reduce} pour utiliser le cluster Hadoop distant à utiliser.
6. Dans les listes \texttt{Distribution} et \texttt{Version}, sélectionnez la distribution de Hadoop que vous utilisez. Dans cet exemple, sélectionnez \textit{HortonWorks Data Platform V2.1.0 (Baikal)}.
7. Dans la liste **Load function**, sélectionnez **PigStorage**. Les paramètres à configurer s’affichent.


9. Cochez la case **Set Resourcemanager scheduler address** et saisissez l’URI du service dans le champ qui s’affiche. Cela vous permet d’utiliser le service d’ordonnancement (Scheduler) défini dans le cluster Hadoop à utiliser. Si ce service n’est pas défini dans votre cluster, ignorez cette étape.


11. Dans le champ **Input file URI**, saisissez le chemin d’accès à la relation de laquelle vous souhaitez lire des données. Comme expliqué précédemment, la relation à lire ici est celle contenant les données des maîtres et de leur animal.

12. Dans le champ **Field separator**, saisissez le séparateur des données à lire. Dans cet exemple, saisissez un point-virgule `;`.

**Charger les données d’exemple relatives aux étudiants et leurs amis**

**Procédure**

1. Double-cliquez sur le second **tPigLoad** pour ouvrir sa vue **Component**.

![Image](image.png)

2. Cliquez sur le bouton `[...]` à côté du champ **Edit schema** pour ouvrir l’éditeur du schéma.

3. Cliquez deux fois sur le bouton `[+]` pour ajouter deux lignes et, dans la colonne **Column**, renommez-les `student` et `friend`, respectivement.
4. Cliquez sur **OK** pour valider ces modifications et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

5. Dans la zone **Mode**, sélectionnez **Map/Reduce**.
   
   Ce composant réutilise la connexion à Hadoop configurée dans le composant principal **tPigLoad**. Les listes **Distribution** et **Version** ont été automatiquement renseignées avec les valeurs du composant principal de chargement.

6. Dans la liste **Load function**, sélectionnez la fonction **PigStorage** pour lire les données source.


**Agréger les relations**

**Procédure**

1. Double-cliquez sur le **tPigCoGroup** pour ouvrir sa vue **Component**.

2. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur de schéma.
3. Cliquez cinq fois sur le bouton [+], pour ajouter cinq lignes et, dans la colonne Column, renommez-les owner_friend, age, pet_number, pet et student, respectivement.

4. Dans la colonne Type de la ligne age, sélectionnez Integer.

5. Cliquez sur OK pour valider ces modifications et acceptez la propagation par la boîte de dialogue qui s’ouvre.

6. Dans la table Group by, cliquez sur le bouton [+], pour ajouter une ligne.

7. Vous devez paramétrer les conditions de groupe dans la table Group by table pour agréger les deux relations d’entrée. Dans chaque colonne représentant la relation d’entrée, cliquez sur la nouvelle ligne et sélectionnez la colonne à utiliser pour composer la condition de groupe. Dans ce scénario, la colonne owner de la relation maître-animal et la colonne friend de la relation étudiant-ami sont sélectionnées car elles ont des enregistrements en commun. A partir de ces colonnes, les deux relations sont agréées dans des bags (entre accolades).

Les bags concernant l’enregistrement Alice se présentent comme suit :

```
Alice, 
(Alice,turtle,17),
(Alice,goldfish,17),
(Alice,cat,17),
(Cindy,Alice),
(Mark,Alice)
```

8. Dans la table Output mapping, le schéma de sortie défini précédemment a automatiquement renseigné la colonne Column. Vous devez renseigner cette table afin de définir comment les bags groupés sont agréés dans le schéma de la relation de sortie. La liste suivante fournit plus de détails concernant la configuration de cette agrégation dans ce scénario :

<table>
<thead>
<tr>
<th>Colonne</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>owner_friend</td>
<td>Reçoit les enregistrements provenant des colonnes utilisées comme condition de groupe. Pour cette raison, sélectionnez la fonction EMPTY dans la liste déroulante Function, pour que les enregistrements entrants restent tels qu’ils sont. Sélectionnez row1 dans la liste Source schema et owner dans la liste Expression pour lire les enregistrements de la colonne d’entrée correspondante. Vous pouvez également sélectionner row2 et friend, les enregistrements à recevoir sont les mêmes car les colonnes owner et friend sont jointes lorsqu’elles sont utilisées comme condition de groupe. Notez que le libellé row1 est l’ID du lien d’entrée et peut être différent dans votre scénario.</td>
</tr>
<tr>
<td>age</td>
<td>Reçoit les données d’âge. Comme dans les bags dans l’étape précédente, l’âge d’un propriétaire apparaît de manière répétée dans l’un des bags après regroupement. Vous pouvez sélectionner la fonction AVG dans la liste Function pour calculer la moyenne des valeurs répétitives,</td>
</tr>
</tbody>
</table>
Colonne | Description
-------- | ---------------------------------
        | comme l’âge qui apparaît une seule fois dans le résultat final. Sélectionnez row1 dans la liste Source schema et age dans la liste Expression.

**pet_number**

Reçoit le nombre d’animaux d’une personne.

Sélectionnez la fonction COUNT dans la liste Function pour effectuer le calcul. Sélectionnez row1 dans la liste Source schema et pet dans la liste Expression.

**pet et student**

Reçoivent les enregistrements groupés des colonnes d’entrée pet et student, respectivement.

Sélectionnez EMPTY pour les deux colonnes et, dans la liste Source schema de chacune, sélectionnez le schéma d’entrée correspondant. Dans la liste Expression, sélectionnez la colonne correspondante.

**Ecrire les données agrégées**

**Procédure**

1. Double-cliquez sur le tPigStoreResult pour ouvrir sa vue Component.

2. Si ce composant n’a pas le même schéma que le composant précédent, une icône d’avertissement s’affiche. Dans ce cas, cliquez sur le bouton Sync columns afin de récupérer le schéma du composant précédent. Cela fait, l’icône disparaît.

3. Dans le champ Result folder URI, saisissez le chemin d’accès dans HDFS pointant vers l’emplacement dans lequel vous souhaitez écrire les résultats.

4. Cochez la case Remove result directory if exists.

5. Dans la liste Store function, sélectionnez PigStorage.
6. Dans le champ Field separator, saisissez le séparateur que vous souhaitez utiliser. Dans ce scénario, saisissez une virgule ",".

**Exécuter le Job**

Appuyez sur F6 pour exécuter le Job.

Cela fait, vérifiez les résultats du système HDFS que vous utilisez.

Vous pouvez voir, par exemple, que Alice a 17 ans, elle possède trois animaux : un chat, un poisson rouge et une tortue. Deux de ses amis s'appellent Mark et Cindy.
tPigCross

Ce composant utilise l'opérateur CROSS pour effectuer un produit cartésien entre deux relations ou plus.
Le tPigCross effectue un produit cartésien entre un fichier d’entrée et un fichier de référence.

Propriétés du tPigCross Standard

Ces propriétés sont utilisées pour configurer le tPigCross s’exécutant dans le framework de Jobs Standard.
Le composant tPigCross Standard appartient aux familles Big Data et Processing.
Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |


<p>| Cross filename | Spécifiez le chemin d’accès du fichier pour le produit cartésien. |</p>
<table>
<thead>
<tr>
<th>Field separator</th>
<th>Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advanced settings</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Use partitioner</strong></td>
<td>Cochez cette case pour spécifier le Partitioner Hadoop qui contrôle le partitionnement des clés des map-sorties intermédiaires. Pour plus d’informations concernant l’utilisation du Partitioner Hadoop, consultez : <a href="http://hadoop.apache.org/docs/r2.2.0/api/org/apache/hadoop/mapred/Partitioner.html">http://hadoop.apache.org/docs/r2.2.0/api/org/apache/hadoop/mapred/Partitioner.html</a> (en anglais).</td>
</tr>
<tr>
<td><strong>Increase parallelism</strong></td>
<td>Cochez cette case pour définir le nombre de tâches &quot;reduce&quot; pour les Jobs MapReduce.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
<tr>
<td><strong>Global Variables</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Global Variables</strong></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option. Une variable <em>Flow</em> fonctionne durant l’exécution d’un composant. Une variable <em>After</em> fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <em>Ctrl+Espace</em> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
<tr>
<td><strong>Utilisation</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Règle d’utilisation</strong></td>
<td>Ce composant est généralement utilisé comme étape intermédiaire. Il nécessite donc un composant d’entrée et un composant de sortie.</td>
</tr>
</tbody>
</table>
| **Prérequis** | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*. La liste suivante présente des informations d’exemple relatives à MapR.  
  * Assurez-vous d’avoir installé le client MapR sur la même machine que le *Studio Talend* et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client |
MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante :
no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

| Limitation | La connaissance des scripts Pig est nécessaire. |

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tPigDistinct

Ce composant supprime des tuples en doublon dans une relation.

**Propriétés du tPigDistinct Standard**

Ces propriétés sont utilisées pour configurer le tPigDistinct s’exécutant dans le framework de Jobs Standard.

Le composant tPigDistinct Standard appartient aux familles Big Data et Processing.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• View schema : sélectionnez cette option afin de voir le schéma.  
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| --- | --- |

**Advanced settings**

| Increase parallelism | Cochez cette case pour définir le nombre de tâches "reduce" pour les Jobs MapReduce. |
tStatCatcher Statistics

Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

Global Variables

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé comme étape intermédiaire. Il nécessite donc un composant d’entrée et un composant de sortie.

⚠️ **Avertissement** :

Ce composant ne conserve pas l’ordre original du fichier d’entrée.

**Prérequis**

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le *Studio Talend* et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées **MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native**. Par exemple, pour Windows, la bibliothèque est **lib\MapRClient.dll** dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : *no MapRClient in java.library.path.*

- Configurez l’argument *-Djava.library.path*, par exemple, dans la zone *Job Run VM arguments*.
La connaissance des scripts Pig est nécessaire.

| Limitation | La connaissance des scripts Pig est nécessaire. |

**Scénario associé**

Pour plus d’informations concernant l’utilisation du composant **tPigDistinct**, consultez le scénario du composant **tPigFilterRow**, dans Scénario : Filtrer des lignes de données selon une condition et sauvegarder le résultat dans un fichier local à la page 3121.
tPigFilterColumns

Ce composant sélectionne des données ou filtre des données d'une relation à partir de conditions de filtre définies.

Le tPigFilterColumns sélectionne une ou plusieurs colonne(s) d'une relation à partir d'une condition définie.

Propriétés du tPigFilterColumns Standard

Ces propriétés sont utilisées pour configurer le tPigFilterColumns s'exécutant dans le framework de Jobs Standard.

Le composant tPigFilterColumns Standard appartient aux familles Big Data et Processing.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |


3116
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme étape intermédiaire. Il nécessite donc un composant d’entrée et un composant de sortie. |

| Prérequis | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend.  

La liste suivante présente des informations d’exemple relatives à MapR.  

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les bibliothèques du client MapR correspondant à chaque OS peuvent être trouvées MAPR\INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).  

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.  

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument |
fournit au studio le chemin d'accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d'utiliser entièrement l'aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

| Limitation         | La connaissance des scripts Pig est nécessaire. |

**Scénario associé**

Pour plus d’informations concernant l’utilisation du composant **tPigFilterColumns**, consultez le scénario du composant **tPigJoin** dans Scénario : Effectuer une jointure sur deux fichiers à partir d’une correspondance exacte et sauvegarder le résultat dans un fichier local à la page 3128.
tPigFilterRow

Ce composant applique des conditions de filtre sur une ou plusieurs colonne(s) spécifiée(s), dans un processus Pig, afin de diviser ou filtrer des données d’une relation.

Le tPigFilterRow filtre ou divise le flux d’entrée d’une séquence Pig selon un ensemble de conditions sur une (des) colonne(s) donnée(s).

Propriétés du tPigFilterRow Standard

Ces propriétés sont utilisées pour configurer le tPigFilterRow s’exécutant dans le framework de Jobs Standard.

Le composant tPigFilterRow Standard appartient aux familles Big Data et Processing.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
  • View schema : sélectionnez cette option afin de voir le schéma.
  • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
  • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |
| Filter configuration | Cliquez sur le bouton Add sous la table Filter configuration pour ajouter une ou plusieurs condition(s) de filtre. |
Notez que lorsque la colonne utilisée pour la condition est de type **String**, le texte saisi dans la colonne **Value** doit être entouré par des guillemets simples et doubles (par exemple “California”). En effet, les guillemets doubles sont requis par le générateur de code de **Talend** et les guillemets simples sont requis par la grammaire de Pig.

**Remarque :**
Cette table disparaît si vous cochez la case **Use advanced filter**.

### Use advanced filter

Cochez cette case afin de définir les conditions avancées de filtrer en saisissant une expression de filtre dans le champ **Filter**.

### Advanced settings

- **tStatCatcher Statistics**

  Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

- **Global Variables**

  **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

  Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

  Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

  Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

- **Règle d’utilisation**

  Ce composant est généralement utilisé comme étape intermédiaire dans une séquence Pig.

- **Prérequis**

  La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le **Studio Talend**. La liste suivante présente des informations d’exemple relatives à MapR.

  - Assurez-vous d’avoir installé le client MapR sur la même machine que le **Studio Talend** et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client
MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est \lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

| Limitation | La connaissance des scripts Pig est nécessaire. |

**Scénario : Filtre des lignes de données selon une condition et sauvegarder le résultat dans un fichier local**

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario décrit un Job à quatre composants filtrant une liste de clients afin de trouver les clients d’un pays spécifique et sauvegardant la liste qui en résulte dans un fichier local. Avant de filtrer les données d’entrée, les entrées en doublon sont supprimées de la liste.

Le fichier d’entrée contient trois colonnes : Name, Country et Age et contient des doublons, comme montré ci-dessous :

| Mario;PuertoRico;49 |
| Mike;USA;22 |
| Ricky;PuertoRico;37 |
| Silvia;Spain;20 |
| Billy;Canada;21 |
| Ricky;PuertoRico;37 |
| Romeo;UK;19 |
| Natasha;Russia;25 |
| Juan;Cuba;23 |
| Bob;Jamaica;55 |
| Mario;PuertoRico;49 |
Déposer et relier les composants

Procédure
1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un tPigLoad, un tPigDistinct, un tPigFilterRow et un tPigStoreResult.
2. Cliquez-droit sur le tPigLoad, sélectionnez Row > Pig Combine dans le menu contextuel et cliquez sur le tPigDistinct pour relier ces deux composants.
3. Répétez cette opération pour relier le tPigDistinct au tPigFilterRow et le tPigFilterRow au tPigStoreResult à l’aide de liens Row > Pig Combine pour former une chaîne Pig.

Configurer les composants

Charger les données d’entrée et supprimer les doublons

Procédure
1. Double-cliquez sur le tPigLoad afin d’ouvrir sa vue Basic settings.
2. Cliquez sur le bouton [...] à côté du champ Edit schema pour ouvrir la boîte de dialogue [Schema].
3. Cliquez sur le bouton [+] pour ajouter trois colonnes, selon la structure du fichier d’entrée : Name (de type String), Country (String) et Age (Integer) puis cliquez OK pour sauvegarder la configuration et fermer la boîte de dialogue.

4. Dans la zone Mode, sélectionnez Local.

5. Dans le champ Input file URI, renseignez le chemin d'accès complet au fichier d’entrée.

6. Sélectionnez PigStorage dans la liste Load function et laissez les autres paramètres tels qu’ils sont.

7. Double-cliquez sur le tPigDistinct pour ouvrir sa vue Basic settings. Cliquez sur le bouton Sync columns pour vous assurer que la structure du schéma d’entrée a bien été propagée depuis le composant précédent.
   Ce composant supprime les doublons du flux d’entrée.

Configurer le filtre

Procédure

1. Double-cliquez sur le tPigFilterRow afin d’ouvrir sa vue Basic settings.

2. Cliquez sur Sync columns pour récupérer la structure du schéma du composant précédent.

3. Cochez la case Use advanced filter et saisissez, dans le champ Filter, l’expression :

   "Country matches 'PuertoRico'"

Cette expression de filtre sélectionne les ligne de données contenant "PuertoRico" dans la colonne Country.
Configurer le fichier de sortie

Procédure
1. Double-cliquez sur le composant tPigStoreResult pour ouvrir sa vue Basic settings.

   ![tPigStoreResult](image)

2. Cliquez sur Sync columns pour récupérer la structure du schéma du composant précédent.
3. Dans le champ Result file, renseignez le chemin d'accès au fichier de résultat.
4. Si le fichier cible existe déjà, cochez la case Remove result directory if exists.
5. Sélectionnez PigStorage dans la liste Store function et laissez les autres paramètres tels qu'il sont.

Sauvegarder et exécuter le Job

Procédure
1. Sauvegardez votre Job en appuyant sur les touches Ctrl+S.
2. Appuyez sur F6 ou cliquez sur le bouton Run de la vue Run pour exécuter le Job.

Résultats

![LuckyCustomers](image)

Le fichier de résultat contient les informations des clients du pays spécifié.
tPigJoin

Ce composant effectue des jointures Inner Join et des jointures Outer Join à partir de clés de jointure de deux fichiers afin de créer des données à utiliser par Pig.

Le tPigJoin vous permet d’effectuer des jointures entre deux fichiers à partir de clés de jointure.

Propriétés du tPigJoin Standard

Ces propriétés sont utilisées pour configurer le tPigJoin s’exécutant dans le framework de Jobs Standard.

Le composant tPigJoin Standard appartient aux familles Big Data et Processing.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Remarque** :

Pour faire fonctionner ce composant, deux schémas doivent être définis : le schéma du flux principal et le schéma du flux de référence. Du côté sortie du schéma principal, les colonnes du fichier d’entrée principal doivent être concaténées manuellement avec celles du fichier de référence.


**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre *Repository Content*.

**Remarque :**

Pour faire fonctionner ce composant, deux schémas doivent être définis : le schéma du flux principal et le schéma du flux de référence. Du côté sortie du schéma principal, les colonnes du fichier d’entrée principal doivent être concaténées manuellement avec celles du fichier de référence.

**Built-in** : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.


**Filename**

Spécifiez le chemin d’accès au fichier *Lookup*.

**Field Separator**

Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.

**Join key**

Cliquez sur le bouton [+ ] pour ajouter des lignes afin de configurer les clés de jointure *Join key* pour les fichiers *Input file* et *Lookup*.

**Join mode**

 Sélectionnez une mode de jointure dans la liste :
**inner-join** : Sélectionnez ce mode afin d’effectuer une jointure Inner Join entre deux relations ou plus, à partir des clés de jointure.

**left-outer-join** : Sélectionnez ce mode afin d’effectuer une jointure Left Outer Join entre deux relations ou plus, à partir de clés de jointure.

**right-outer-join** : Sélectionnez ce mode afin d’effectuer une jointure Right Outer Join entre deux relations ou plus, à partir des clés de jointure.

**full-outer-join** : Sélectionnez ce mode afin de combiner les effets des jointures Left et Right Outer Joins.

Pour plus d’informations concernant les jointures Inner Join et Outer Join, consultez :


---

### Advanced settings

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Optimize the join**   | Cochez cette case afin d’optimiser les performances des jointures à l’aide des jointures REPLICATED, SKewed, ou MERGE. Pour plus d’informations concernant les jointures optimisées, consultez :

http://pig.apache.org/docs/r0.8.1/piglatin_ref1.html#Specialized+Joins (en anglais). |

| **Use partitioner**     | Cochez cette case afin de spécifier le Partitioner Hadoop qui contrôle le partitionnement des clés des map-sorties intermédiaires. Pour plus d’informations concernant l’utilisation du Partitioner Hadoop, consultez :

http://hadoop.apache.org/docs/r2.2.0/api/org/apache/hadoop/mapred/Partitioner.html (en anglais). |

| **Increase parallelism**| Cochez cette case pour définir le nombre de tâches “reduce” pour des Jobs MapReduce. |

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

---

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette</td>
</tr>
<tr>
<td><strong>Règle d'utilisation</strong></td>
<td>Ce composant est généralement utilisé comme étape intermédiaire. Il nécessite donc un composant d’entrée et un composant de sortie.</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Prérequis</strong></td>
<td>La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le <strong>Studio Talend</strong>. La liste suivante présente des informations d’exemple relatives à MapR.</td>
</tr>
<tr>
<td></td>
<td>• Assurez-vous d’avoir installé le client MapR sur la même machine que le <strong>Studio Talend</strong> et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées <code>MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native</code>. Par exemple, pour Windows, la bibliothèque est <code>lib\MapRClient.dll</code> dans le fichier <code>jar</code> du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : <a href="http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr">http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr</a> (en anglais).</td>
</tr>
<tr>
<td></td>
<td>Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : <code>no MapRClient in java.library.path</code>.</td>
</tr>
<tr>
<td></td>
<td>• Configurez l’argument <code>-Djava.library.path</code>, par exemple, dans la zone <strong>Job Run VM arguments</strong> de la vue <strong>Run/Debug</strong> de la boîte de dialogue <strong>[Preferences]</strong> dans le menu <strong>Window</strong>. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (<strong>Data viewer</strong>) afin de visualiser localement dans le studio les données stockées dans MapR.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.</td>
</tr>
<tr>
<td><strong>Limitation</strong></td>
<td>La connaissance des scripts Pig est nécessaire.</td>
</tr>
</tbody>
</table>

**Scénario : Effectuer une jointure sur deux fichiers à partir d’une correspondance exacte et sauvegarder le résultat dans un fichier local**

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.
Ce scénario décrit un Job à quatre composants combinant les données d'un fichier d'entrée et d'un fichier de référence correspondant à une clé de jointure donnée, supprime les colonnes indésirables puis sauvegarde le résultat final dans un fichier local.

Le fichier d'entrée principal contient les informations concernant des personnes : ID, prénom, nom de famille, ID de groupe et salaire, comme ci-dessous :

<table>
<thead>
<tr>
<th>ID</th>
<th>Prénom</th>
<th>Nom de famille</th>
<th>ID de groupe</th>
<th>Salaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Woodrow</td>
<td>Johnson</td>
<td>3</td>
<td>1013.39</td>
</tr>
<tr>
<td>2</td>
<td>Millard</td>
<td>Monroe</td>
<td>2</td>
<td>8077.59</td>
</tr>
<tr>
<td>3</td>
<td>Calvin</td>
<td>Eisenhower</td>
<td>3</td>
<td>6866.88</td>
</tr>
<tr>
<td>4</td>
<td>Lyndon</td>
<td>Wilson</td>
<td>3</td>
<td>5726.28</td>
</tr>
<tr>
<td>5</td>
<td>Ronald</td>
<td>Garfield</td>
<td>2</td>
<td>4158.58</td>
</tr>
<tr>
<td>6</td>
<td>Rutherford</td>
<td>Buchanan</td>
<td>3</td>
<td>2897.00</td>
</tr>
<tr>
<td>7</td>
<td>Calvin</td>
<td>Coolidge</td>
<td>1</td>
<td>6650.66</td>
</tr>
<tr>
<td>8</td>
<td>Ulysses</td>
<td>Roosevelt</td>
<td>2</td>
<td>7854.78</td>
</tr>
<tr>
<td>9</td>
<td>Grover</td>
<td>Tyler</td>
<td>1</td>
<td>5226.88</td>
</tr>
<tr>
<td>10</td>
<td>Bill</td>
<td>Tyler</td>
<td>2</td>
<td>8964.66</td>
</tr>
</tbody>
</table>

Le fichier de référence contient uniquement les informations des IDs de groupes ainsi que le nom des groupes :

<table>
<thead>
<tr>
<th>ID</th>
<th>Nom du groupe</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>group_A</td>
</tr>
<tr>
<td>2</td>
<td>group_B</td>
</tr>
</tbody>
</table>

**Déposer et relier les composants**

**Procédure**

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : tPigLoad, tPigJoin, tPigFilterColumns et tPigStoreResult.
2. Connectez ces composants à l'aide de liens Row > Pig Combine.

**Configurer les composants**

**Charger le fichier d’entrée principal**

**Procédure**

1. Double-cliquez sur le tPigLoad pour ouvrir sa vue Basic settings.
2. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir la boîte de dialogue [Schema].

3. Cliquez sur le bouton [+] pour ajouter des colonnes. Renommez-les et définissez leur type selon la structure du fichier d'entrée. Dans cet exemple, le schéma d'entrée contient cinq colonnes : `id` (integer), `firstName` (string), `lastName` (string), `groupId` (integer) et `salary` (double). Cliquez sur **OK** pour valider et fermer la boîte de dialogue.

4. Cliquez sur **Local** dans la zone **Mode**.

5. Sélectionnez **PigStorage** dans la liste **Load function**.

6. Renseignez le champ **Input file URI** avec le chemin d'accès complet au fichier d'entrée et laissez les autres paramètres tels qu’ils sont.

**Charger le fichier de référence et configurer la jointure Inner Join**

**Procédure**

1. Double-cliquez sur le **tPigJoin** pour ouvrir sa vue **Basic settings**.
2. Cliquez sur le bouton [...] du schéma principal pour ouvrir la boîte de dialogue [Schema].

3. Vérifiez que le schéma d’entrée a bien été récupéré du composant précédent. Si nécessaire, cliquez sur le bouton [-->] pour copier toutes les colonnes du schéma d’entrée au schéma de sortie.

4. Cliquez sur le bouton [+] sous la table de sortie pour ajouter de nouvelles colonnes, selon la structure des données du fichier de référence, groupId_ref (integer) et groupName (string) dans cet exemple. Cliquez sur OK pour fermer la boîte de dialogue.

5. Cliquez sur le bouton [...] du schéma du flux de référence pour ouvrir la boîte de dialogue [Schema].
6. Cliquez sur le bouton [+] sous la table de sortie pour ajouter : groupId_ref (integer) et groupName (string) puis cliquez sur OK pour fermer la boîte de dialogue.

7. Dans le champ Filename, spécifiez le chemin d'accès complet au fichier de référence.

8. Cliquez sur le bouton [+] sous la table Join key pour ajouter une ligne et sélectionnez groupId et groupId_ref, respectivement dans les liste Input et Lookup afin de mettre en correspondance les données du flux d'entrée principal et celles du flux de référence, selon l'ID du groupe.


Définir le schéma de sortie final et le fichier de sortie

Procédure

1. Double-cliquez sur le composant tPigFilterColumns pour ouvrir sa vue Basic settings.

2. Cliquez sur le bouton [...] à côté du champ Edit schema pour ouvrir la boîte de dialogue [Schema].
3. Dans le schéma d’entrée, sélectionnez les colonnes à inclure dans le fichier de résultat en cliquant sur celles-ci une par une en maintenant la touche \textit{Ctrl} enfoncée. Cliquez sur le bouton [-] pour les copier dans le schéma de sortie. Cliquez sur \textit{OK} pour valider le schéma et fermer la boîte de dialogue.

Dans cet exemple, le fichier de résultat doit inclure toutes les informations, sauf l’ID des groupes.

4. Double-cliquez sur le \texttt{tPigStoreResult} pour afficher sa vue \textit{Basic settings}.

5. Cliquez sur \textit{Sync columns} afin de récupérer la structure du schéma du composant précédent.

6. Dans le champ \textit{Result file}, saisissez le chemin d’accès complet au fichier de résultat et cochez la case \textit{Remove result file directory if exists}.

7. Sélectionnez \texttt{PigStorage} dans la liste \textit{Store function} et laissez les autres paramètres tels qu’ils sont.

\textbf{Sauvegarder et exécuter le Job}

\textbf{Procédure}

1. Appuyez sur les touches \texttt{Ctrl+S} pour sauvegarder votre Job.

2. Appuyez sur \texttt{F6} ou cliquez sur le bouton \texttt{Run} dans la vue \texttt{Run} pour exécuter le Job.

Le fichier de résultat contient toutes les informations relatives aux personnes des groupes A et B, sauf l’ID des groupes.
<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>Value</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Calvin</td>
<td>6650.66</td>
<td>group_A</td>
</tr>
<tr>
<td>2</td>
<td>Grover</td>
<td>5226.88</td>
<td>group_A</td>
</tr>
<tr>
<td>3</td>
<td>Millard</td>
<td>5077.59</td>
<td>group_B</td>
</tr>
<tr>
<td>4</td>
<td>Ronald</td>
<td>4155.58</td>
<td>group_B</td>
</tr>
<tr>
<td>5</td>
<td>Ulysses</td>
<td>7654.78</td>
<td>group_B</td>
</tr>
<tr>
<td>6</td>
<td>Bill</td>
<td>8964.66</td>
<td>group_B</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
tPigLoad

Ce composant charge les données originales d’entrée dans un flux de sortie en une seule transaction, une fois que les données ont été validées.

Le tPigLoad établit une connexion à la source des données pour la transaction courante.

Propriétés du tPigLoad Standard

Ces propriétés sont utilisées pour configurer le tPigLoad s’exécutant dans le framework de Jobs Standard.

Le composant tPigLoad Standard appartient aux familles Big Data et Processing.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>: propriétés utilisées ponctuellement.</td>
</tr>
</tbody>
</table>

Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la
<table>
<thead>
<tr>
<th>Local</th>
<th>Cliquez sur ce bouton radio afin d'exécuter des scripts Pig en mode Local. Dans ce mode, tous les fichiers sont installés et exécutés à partir de votre hôte et de votre système de fichiers locaux.</th>
</tr>
</thead>
</table>
| Tez | Sélectionnez cette option pour exécuter le Job Pig avec le framework Tez. Ce mode Tez est disponible uniquement lorsque vous utilisez l'une des distributions suivantes :  
- Hortonworks : V2.2 +.  
- Custom : cette option vous permet de vous connecter à une distribution supportant Tez mais non officiellement supportée par Talend.  
Avant d'utiliser Tez, vérifiez que votre cluster Hadoop supporte Tez. Vous devez configurer l'accès aux bibliothèques Tez correspondantes via la vue Advanced settings de ce composant. Pour plus d'informations concernant Pig avec Tez, consultez la documentation Apache à l'adresse suivante : https://cwiki.apache.org/confluence/display/PIG/Pig+on+Tez (en anglais). |
| Map/Reduce | Cliquez sur ce bouton radio afin d'exécuter des scripts Pig en mode Map/Reduce. Une fois ce mode sélectionné, vous devez renseigner les champs dans la zone Configuration qui apparaît :  
- Distribution et Version:  
  Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.  
Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d'informations concernant la configuration d'un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

L’option Custom vous permet de vous connecter à un cluster différente des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.
• **Use Kerberos authentication:**

Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

• Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans Connexion sécurisée à MapR à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

De plus, comme ce composant effectue des calculs Map/Reduce, vous devez également authentifier les services associés, comme le serveur de l’historique des Jobs et le gestionnaire de ressources ou le JobTracker, selon votre distribution, dans le champ correspondant. Ces principaux se trouvent dans les fichiers de configuration de votre distribution. Par exemple, dans une distribution CDH4, le principal du gestionnaire de ressource est configuré dans le fichier `yarn-site.xml` et le principal de l’historique des Job dans le fichier `mapred-site.xml`.

Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

Les principaux relatifs à HBase ne sont requis que pour la fonction **HBaseStorage**.

• **Use a keytab to authenticate:**

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est **user1** et le principal à utiliser est **guest**. Dans
cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser.

- **NameNode URI:**
  Saisissez l’emplacement du NameNode correspondant à la version de Map/Reduce à utiliser. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un t.LibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

- **JobTracker host:**
  Saisissez l’emplacement du ResourceManager correspondant à la version de Map/Reduce à utiliser.

Dans le JobHistory, vous pouvez facilement trouver le statut d’exécution de votre Job Pig car il est automatiquement nommé. Son nom est le résultat de la concaténation du nom du projet contenant le Job, du nom du Job, de sa version et du libellé du premier composant t.PigLoad utilisé. La convention de nommage d’un Job Pig dans le JobHistory est la suivante : ProjectName_JobNameVersion_FirstComponentName.

Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d’un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du de l’exécution) :

1. Cochez la case **Set resourcemanager scheduler address** et saisissez l’adresse de l’ordonnanceur (Scheduler) dans le champ qui apparaît.

2. Cochez la case **Set jobhistory address** et saisissez l’emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory.

3. Cochez la case **Set staging directory** et saisissez le chemin d’accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l’exécution de programmes. Ce répertoire se trouve sous la propriété yarn.app.mapreduce.am.staging-dir dans les fichiers de configuration, notamment les fichiers yarn-site.xml et mapred-site.xml de votre distribution.

4. Allouez des volumes de mémoire aux calculs Map et Reduce et au service ApplicationMaster de YARN en cochant la case **Set memory** dans la vue **Advanced settings**.

5. Cochez la case **Set Hadoop user** et saisissez le nom de l’utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les
droits appropriés de lecture ou d'écriture, ce champ vous permet d'exécuter le Job directement avec l'utilisateur ayant les droits d'accès appropriés au fichier ou répertoire à traiter.

6. Cochez la case **Use datanode hostname** pour permettre au Job d'accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété `dfs.client.use.datanode.hostname` à `true`. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.

- **User name:**
  
  Saisissez le nom de l'utilisateur avec lequel vous souhaitez exécuter le Job. Puisqu'un fichier ou un répertoire dans Hadoop a son auteur spécifique, avec les droits en lecture ou écriture appropriés, ce champ vous permet d’exécuter le Job directement sous le nom d'utilisateur ayant les droits appropriés pour accéder au fichier ou au répertoire à traiter. Notez que ce champ peut n'être pas disponible selon la distribution que vous utilisez.

### WebHCat configuration


Dans le champ **Job result folder**, saisissez l'emplacement où vous souhaitez stocker les résultats d’exécution du Job dans Azure Storage.

### HDInsight configuration

Saisissez les informations d’authentification du cluster HD Insight à utiliser.

### Windows Azure Storage configuration


Dans le champ **Container**, saisissez le nom du conteneur à utiliser.

Dans le champ **Deployment Blob**, saisissez l'emplacement où vous souhaitez stocker le Job et ses bibliothèques dépendantes dans le compte Azure Storage.

### Inspect the classpath for configurations

Cochez cette case pour permettre au composant de vérifier les fichiers de configuration dans le répertoire configuré pour la variable `$HADOOP_CONF_DIR` et de lire directement les paramètres de ces fichiers dans le répertoire. Cette fonctionnalité vous permet de modifier facilement la configuration Hadoop afin que le composant puisse passer d’un environnement à un autre,
comme par exemple pour passer d’un environnement test à un environnement production.

Dans ce cas, les champs ou les options utilisée pour configurer la connexion Hadoop et/ou la sécurité Kerberos sont masqués.

Si vous souhaitez utiliser certains paramètres comme les paramètres Kerberos mais que ces paramètres ne sont pas inclus dans les fichiers de configuration Hadoop, vous devez créer un fichier appelé talend-site.xml et mettre ce fichier dans le répertoire défini dans $HADOOP_CONF_DIR. Le fichier talend-site.xml doit se présenter comme suit :

```xml
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>talend.kerberos.authentication</name>
    <value>kinit</value>
    <description>Set the Kerberos authentication method to use. Valid values are: kinit or keytab. </description>
  </property>
  <property>
    <name>talend.kerberos.keytab.principal</name>
    <value>user@BIGDATA.COM</value>
    <description>Set the keytab's principal name. </description>
  </property>
  <property>
    <name>talend.kerberos.keytab.path</name>
    <value>/kdc/user.keytab</value>
    <description>Set the keytab's path. </description>
  </property>
  <property>
    <name>talend.encryption</name>
    <value>none</value>
    <description>Set the encryption method to use. Valid values are: none or ssl. </description>
  </property>
  <property>
    <name>ssl.trustStore.path</name>
    <value>ssl</value>
    <description>Set SSL trust store path. </description>
  </property>
</configuration>
```
Les paramètres lus depuis ces fichiers de configuration écrasent ceux utilisés par défaut dans le Studio. Lorsqu'un paramètre n'existe pas dans ces fichiers de configuration, le paramètre par défaut est utilisé.

### Load function

 Sélectionnez une fonction dans la liste afin de déterminer comment charger les données.

- **PigStorage** : Charge les données au format UTF-8.
- **BinStorage** : Charge les données dans un format lisible par les machines.
- **TextLoader** : Charge des données non structurées au format UTF-8.
- **HCatLoader** : Charge des données à partir de tables gérées par HCatalog à l'aide de scripts Pig.
- Cette fonction est disponible uniquement lorsque vous avez sélectionné Hortonworks comme distribution Hadoop à utiliser, dans les listes Distribution et Version affichées en mode Map/Reduce. Pour plus d'informations concernant l'utilisation de HCatLoader, consultez [http://hive.apache.org/javadocs/hcat-r0.5.0/api/org/apache/hcatalog/pig/HCatLoader.html](http://hive.apache.org/javadocs/hcat-r0.5.0/api/org/apache/hcatalog/pig/HCatLoader.html) (en anglais).
- **HBaseStorage** : Charge des données de HBase. Vous devez terminer la configuration de HBase dans la zone **HBase configuration** affichée.
- **SequenceFileLoader** : Charge des données au format SequenceFile. Vous devez terminer la configuration du fichier à charger dans la zone **Sequence Loader Configuration** qui s'affiche. Cette fonction ne concerne que le mode Map/Reduce.
- **RCFilePigStorage** : Charge des données au format RCFile format. Cette fonction ne concerne que le mode Map/Reduce.
- **AvroStorage** : Charge des fichiers Avro. Pour plus d'informations concernant AvroStorage, consultez la documentation Apache sur le site [https://cwiki.apache.org/confluence/display/PIG/AvroStorage](https://cwiki.apache.org/confluence/display/PIG/AvroStorage) (en anglais). Cette fonction concerne uniquement le mode Map/Reduce.
- **ParquetLoader** : Charge le fichier Parquet. Fonctionne uniquement en mode Map/Reduce.
- **Custom** : Charge les données à l'aide d'une fonction de chargement personnalisée. Pour ce faire, vous devez enregistrer, dans l'onglet **Advanced settings**, le fichier Jar contenant la fonction à utiliser, puis,
Dans le champ affiché à côté du champ **Load function**, spécifier cette fonction.

Par exemple, après avoir enregistré un fichier Jar nommé `piggybank.jar`, vous pouvez saisir `org.apache.pig.piggybank.storage.XMLLoader('attr') as (xml:chararray)` pour utiliser la fonction personnalisée `XMLLoader`, contenue dans ce fichier Jar. Pour plus d'informations concernant ce Jar `piggybank.jar`, consultez [https://cwiki.apache.org/confluence/display/PIG/PiggyBank](https://cwiki.apache.org/confluence/display/PIG/PiggyBank) (en anglais).

Notez que, lorsque le format de fichier à utiliser est **PARQUET**, il est possible qu'il vous soit demandé de trouver le fichier Jar Parquet spécifique et l'installer dans le studio.

- Lorsque le mode de connexion à Hive est **Embedded**, le Job est exécuté sur votre machine locale et appelle ce Jar installé dans le studio.
- Lorsque le mode de connexion à Hive est **Standalone**, le Job est exécuté sur le serveur hébergeant Hive et ce fichier Jar est envoyé au système HDFS du cluster auquel vous vous connectez. Assurez-vous d'avoir correctement défini l'URI du NameNode dans le champ correspondant de la vue **Basic settings**.

Ce fichier Jar est téléchargeable depuis le site Web d'Apache. Vous pouvez trouver plus d'informations concernant l'installation des modules externes dans Talend Help Center ([https://help.talend.com](https://help.talend.com)).

<table>
<thead>
<tr>
<th>Input file URI</th>
<th>Saisissez dans ce champ le chemin d'accès local complet au fichier d'entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remarque :</strong></td>
<td>Ce champ est indisponible lorsque vous sélectionnez HCatLoader dans la liste Load function ou lorsque vous utilisez un endpoint S3.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use S3 endpoint</th>
<th>Cochez cette case pour lire des données d'un dossier d'un bucket Amazon S3.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Après avoir coché la case <strong>Use S3 endpoint</strong>, vous devez renseigner les paramètres suivants dans les champs qui apparaissent :</td>
<td></td>
</tr>
<tr>
<td><strong>S3 bucket name and folder</strong> : saisissez le nom du bucket et celui du dossier à partir duquel vous souhaitez lire les données. Vous devez séparer le nom du bucket et le nom du dossier par une barre oblique (/).</td>
<td></td>
</tr>
<tr>
<td><strong>Access key</strong> et <strong>Secret key</strong> : renseignez les informations d'authentification requises pour vous connecter au bucket Amazon S3 à utiliser.</td>
<td></td>
</tr>
<tr>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe</td>
<td></td>
</tr>
</tbody>
</table>
entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

Notez que le format du fichier S3 est S3N (S3 Native Filesystem).

<table>
<thead>
<tr>
<th>HCatalog Configuration</th>
</tr>
</thead>
</table>
| Renseignez les champs suivants pour configurer les tables gérées de HCatalog dans HDFS (système de fichiers distribué Hadoop) :

**Distribution et Version** :

 Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option **Custom** vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**.


**Database** : Base de données dans laquelle les tables sont stockées.

**Table** : Table dans laquelle les données sont stockées.

**Partition filter** : Renseignez ce champ avec les clés de partitions afin de lister les partitions par filtre.

**Remarque** :

### Field separator

Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.

*Remarque :*

Ce champ est activé uniquement lorsque vous sélectionnez **PigStorage** dans la liste **Store function**.

### Compression

Cochez la case **Force to compress the output data** afin de compresser les données lorsqu’elles sont écrites en sortie par un **tPigStoreResult**, à la fin d’un processus Pig.

Hadoop fournit différents formats de compression permettant de réduire l’espace nécessaire au stockage des fichiers et d’accélérer le transfert de données. Lorsque vous écrivez un fichier et compressez les données à l’aide du programme Pig, vous devez par défaut ajouter un format de compression en tant que suffixe au chemin d’accès pointant vers le dossier dans lequel vous souhaitez écrire des données, par exemple `/user/ychen/out.bz2`. Cependant, si vous cochez cette case, les données de sortie seront compressées, même si vous n’ajoutez pas de format de compression au chemin d’accès, par exemple `/user/ychen/out`.

*Remarque :*

Le chemin de sortie est configuré dans la vue **Basic settings** du composant **tPigStoreResult**.

### HBase configuration

Cette zone est disponible pour la fonction **HBaseStorage**.

Les paramètres à configurer sont :

**Zookeeper quorum** :

Saisissez le nom ou l’URL du service Zookeeper utilisé pour coordonner les transactions entre votre **Studio** et votre base de données. Notez que, lorsque vous configurez Zookeeper, vous pouvez avoir besoin de configurer explicitement la propriété `zookeeper.znode.parent` pour définir le chemin vers le nœud znode racine contenant tous les znodes créés et utilisés par votre base de données. Cochez la case la case **Set Zookeeper znode parent** afin de définir cette propriété.

**Zookeeper client port** :

Saisissez le numéro du port d’écoute client du service Zookeeper que vous utilisez.

**Table name** :

Saisissez le nom de la table HBase à partir de laquelle vous souhaitez charger des données.

**Load key** :

Cochez cette case pour charger la colonne Row Key comme première colonne du schéma de résultat. Dans cette situation, vous devez avoir créé cette colonne dans le schéma.

**Mapping** :
| **Sequence Loader configuration** | Cette zone est disponible uniquement pour la fonction **SequenceFileLoader**. Comme un enregistrement SequenceFile se compose de paires binaires clé/valeur, les paramètres à configurer sont les suivants :

- **Key column**: Sélectionnez la colonne clé d’un enregistrement clé/valeur.
- **Value column**: Sélectionnez la colonne de valeur d’un enregistrement clé/valeur.
 |

| **Die on subjob error** | Cette case est cochée par défaut, pour ignorer les lignes en erreur dans le sous-job et terminer le traitement des lignes sans erreur. |

### Advanced settings

| **Tez lib** | Choisissez comment accéder aux bibliothèques de Tez :

- **Auto install**: lors de l’exécution, le Job charge et déploie les bibliothèques de Tez fournies par le Studio dans le répertoire spécifié dans le champ **Install folder in HDFS**, par exemple, `/tmp/usr/tez`.

  Si vous avez configuré la propriété `tez.lib.uris` dans la table des propriétés, ce répertoire écrase la valeur de la propriété lors de l’exécution. Les autres propriétés configurées dans la table sont toujours prises en compte.

- **Use exist**: le Job accède aux bibliothèques de Tez déjà déployées dans le cluster Hadoop à utiliser. Vous devez saisir le chemin d’accès pointant vers ces bibliothèques dans le champ **Lib path (folder or file)**.

- **Lib jar**: cette table s’affiche lorsque vous avez sélectionné **Auto install** dans la liste **Tez lib** et que vous utilisez une distribution personnalisée (**Custom**). Dans cette table, vous devez ajouter les bibliothèques de Tez à charger.
 |

| **Hadoop Properties** | Le **Studio Talend** utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- **Notez que**, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**. |
Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :


### Register jar

Cliquez sur le bouton [+] pour ajouter des lignes à la table et, dans ces lignes, parcourez votre système jusqu’aux fichiers Jar à ajouter. Par exemple, pour enregistrer un fichier Jar nommé `piggybank.jar`, cliquez sur le bouton [+] pour ajouter une ligne, puis cliquez sur cette ligne pour afficher le bouton [...] (permettant de parcourir votre système) puis cliquez sur ce bouton pour parcourir votre système jusqu’au fichier `piggybank.jar`, dans l’assistant [Select Module] qui s’ouvre.

### Define functions

Utilisez cette table pour définir les fonctions personnalisées (UDF, [User-Defined Functions]), particulièrement celles nécessitant un alias, comme les fonctions Apache DataFu Pig, à exécuter lors du chargement des données.

Cliquez sur le bouton pour ajouter autant de lignes que nécessaire. Vous devez spécifier un alias et une fonction personnalisée dans les champs correspondants, pour chaque ligne.

Si votre Job comprend un composant `tPigMap`, une fois définies les fonctions personnalisées pour ce composant dans le `tPigMap`, la table est automatiquement renseignée. De la même manière, une fois les fonctions personnalisées définies dans cette table, la table Define functions de l’éditeur de mapping du `tPigMap` est automatiquement renseignée.

Pour plus d’informations concernant la définition de fonctions personnalisées lors du mapping de flux Pig, consultez la section relative au mapping de flux de Big Data, dans le [Guide de prise en main de Talend Open Studio for Big Data](https://www.talend.com/products/big-data/)


### Pig properties

Le [Studio Talend](https://www.talend.com/products/big-data/) utilise la configuration par défaut pour son moteur Pig affin d’effectuer des opérations. Si vous...
devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.
Par exemple, la clé `default_parallel` utilisée dans Pig peut être 20.

| HBaseStorage configuration | Ajoutez et configurez plus d’options de chargement HBaseStorage dans cette table. Les options sont :
|:--------------------------|:-------------------------------------------------------------------|
| `gt` | la valeur minimale de clé, |
| `lt` | la valeur maximale de clé, |
| `gte` | la valeur minimale de clé (comprise), |
| `lte` | la valeur maximale de clé (comprise), |
| `limit` | nombre maximum de lignes à récupérer par région, |
| `caching` | nombre de lignes à mettre en cache, |
| `caster` | le convertisseur à utiliser pour lire des valeurs de HBase, par exemple HBaseBinaryConverter. |

| Define the jars to register for HCatalog | Cette case apparaît lorsque vous utilisez `tHCatLoader`, mais vous pouvez la laisser décochée, car le `Studio Talend` enregistre automatiquement les fichiers Jar requis. Si le fichier Jar est manquant, vous pouvez cocher cette case pour afficher la table `Register jar for HCatalog` et configurer le chemin d’accès à ce Jar manquant. |

| Path separator in server | Laissez le champ `Path separator in server` tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d’autres termes, changez le séparateur si celui-ci n’est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte. |

| Mapred job map memory mb et Mapred job reduce memory mb | Vous pouvez personnaliser les opérations map et reduce en cochant la case `Set memory`, pour configurer les allocations de mémoire pour ces opérations à effectuer par le système Hadoop.
Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations map et reduce dans les champs `Mapred job map memory mb` et `Mapred job reduce memory mb`, respectivement. Par défaut, les valeurs sont toutes les deux 1000, ce qui est normalement adapté pour l’exécution de ces opérations.
Les paramètres de mémoire à définir sont `Map (in Mb)`, `Reduce (in Mb)` et `ApplicationMaster (in Mb)`. Ces champs permettent d’allouer dynamiquement de la mémoire aux opérations map et reduce et à l’ApplicationMaster de YARN. |
**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

---

**Global Variables**

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

---

**Utilisation**

**Règle d’utilisation**

Ce composant est utilisé pour démarrer une séquence Pig et requiert un *tPigStoreResult* en fin de séquence pour écrire les données en sortie.

En mode Map/Reduce, vous devez configurer uniquement la connexion Hadoop pour le premier composant *tPigLoad* d’un processus (sous-job). Tout autre *tPigLoad* dans ce processus réutilise automatiquement la connexion créée par le premier *tPigLoad*.

**Prérequis**

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le *Studio Talend*. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le *Studio Talend* et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les bibliothèques du client MapR correspondant à chaque OS peuvent être trouvées *MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native*. Par exemple, pour Windows, la bibliothèque est *lib\MapRClient.dll* dans le fichier *jar* du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : *no MapRClient in java.library.path.*
**Scénario : Charger une table HBase**

Ce scénario s'applique uniquement aux solutions Talend avec Big Data.

Ce scénario utilise les composants **tPigLoad** et **tPigStoreResult** pour lire des données de HBase et les écrire dans HDFS.

La table HBase à utiliser contient trois colonnes : *id*, *name* et *age*. *id* et *age* appartiennent à la famille de colonnes, *family1* et *name* appartient à la famille de colonnes *family2*.

Les données stockées dans la table HBase se présentent comme suit :

```
1;Albert;23
2;Alexandre;24
3;Alfred-Hubert;22
4;Andre;40
5;Didier;28
6;Anthony;35
7;Artus;32
8;Catherine;34
9;Charles;21
10;Christophe;36
11;Christian;67
12;Danniel;54
13;Elisabeth;58
14;Emile;32
15;Gregory;30
```
Pour reproduire ce scénario, effectuez les opérations suivantes :

**Relier les composants**

**Procédure**

1. Dans la perspective **Integration** du *Studio Talend*, créez un Job vide depuis le nœud **Job Designs** du *Repository* et nommez-le, par exemple, *hbase_storage*.
   
   Pour plus d’informations concernant la création d’un Job, consultez le *Guide utilisateur du Studio Talend*.

2. Déposez un *tPigLoad* et un *tPigStoreResult* dans l’espace de modélisation graphique.

3. Reliez-les à l’aide d’un lien *Row > Pig combine*.

**Configurer le tPigLoad**

**Procédure**

1. Double-cliquez sur le *tPigLoad* pour ouvrir sa vue **Component**.
2. Cliquez sur le bouton \(\text{Edit schema}\) à côté du champ \text{Edit schema} pour ouvrir l’éditeur de schéma.

3. Cliquez quatre fois sur le bouton \(\text{Add rows}\) pour ajouter quatre lignes et renommez-les respectivement : rowkey, id, name et age. La colonne rowkey est la première du schéma, afin de stocker la colonne Row key de HBase, mais si vous n’avez pas besoin de charger cette colonne Row key, vous pouvez créer uniquement les trois autres colonnes du schéma.
4. Cliquez sur OK pour valider les modifications et acceptez la propagation lorsqu’une boîte de dialogue vous la propose.

5. Dans la zone Mode, sélectionnez Map/Reduce, puisque, dans cet exemple, vous utilisez une distribution Hadoop distante.


7. Dans le champ Load function, sélectionnez HBaseStorage. Les paramètres à configurer apparaissent.

8. Dans les champs NameNode URI et JobTracker host, saisissez l’emplacement de ces services. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.


10. Si l’emplacement de Zookeeper znode parent a été défini dans le cluster Hadoop auquel vous vous connectez, cochez la case Set zookeeper znode parent et saisissez la valeur de la propriété dans le champ affiché.

11. Dans le champ Table name, saisissez le nom de la table dont le tPigLoad lit les données.

12. Cochez la case Load key si vous devez charger la colonne Row key de HBase. Dans cet exemple, la case est cochée.

13. Dans la table Mapping, quatre lignes ont été automatiquement ajoutées. Dans la colonne Column family:qualifier, saisissez le nom des colonnes HBase en face des colonnes du schéma défini. Dans ce scénario, saisissez family1:id pour la colonne id, family2:name pour la colonne name et family1:age pour la colonne age.

Configurer le tPigStoreResult

Procédure

1. Double-cliquez sur le tPigStoreResult pour ouvrir sa vue Component.
2. Dans le champ **Result file**, saisissez le chemin d'accès au répertoire dans lequel stocker le résultat. Comme le **tPigStoreResult** réutilise automatiquement la connexion créée par le **tPigLoad**, le chemin d'accès dans ce scénario est le répertoire de la machine hébergeant la distribution Hadoop à utiliser.

3. Cochez la case **Remove result directory if exists**.

4. Dans le champ **Store function**, sélectionnez **PigStorage** pour stocker les résultats au format UTF-8.

**Exécuter le Job**

Appuyez sur **F6** pour exécuter le Job.

Vous pouvez vérifier les résultats dans le système HDFS utilisé.

File: `/user/ychen/sort/customer/part-m-00000`

Si vous souhaitez plus d'informations concernant le Job et son exécution, utilisez la console Web fournie par la distribution Hadoop que vous utilisez.

Dans le JobHistory, vous pouvez facilement trouver le statut d'exécution de votre Job Pig car il est automatiquement nommé. Son nom est le résultat de la concaténation du nom du projet contenant le Job, du nom du Job, de sa version et du libellé du premier composant **tPigLoad** utilisé. La convention de nommage d'un Job Pig dans le JobHistory est la suivante: **ProjectName_JobNameVersion_FirstComponentName**.
**tPigMap**

Ce composant transforme et route des données d’une ou plusieurs source(s) et vers une ou plusieurs destination(s).

Le tPigMap transforme et route les données dans un processus Pig. Il fournit une interface graphique permettant de configurer de manière sophistiquée de multiples flux de données.

### Paramètres de mapping facultatifs

**Du côté de l’entrée :**

<table>
<thead>
<tr>
<th>Propriétés du flux de référence (Lookup)</th>
<th>Valeur</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Join Model</em></td>
<td>Inner Join ; Left Outer Join ; Right Outer Join ; Full Outer Join.</td>
</tr>
<tr>
<td><em>Join Optimization</em></td>
<td>None ; Replicated ; Skewed ; Merge.</td>
</tr>
<tr>
<td><em>Custom Partitioner</em></td>
<td>Saisissez le Partitioner Hadoop à utiliser pour contrôler le partitionnement des clés des map-sorties intermédiaires. Par exemple, saisissez entre guillemets : `org.apache.pig.test.utils.SimpleCustomPartitioner` pour utiliser le Partitioner <em>SimpleCustomPartitioner</em>. Pour plus d’informations concernant le code de ce <em>SimpleCustomPartitioner</em>, consultez le manuel de référence de Pig Latin.</td>
</tr>
<tr>
<td><em>Increase Parallelism</em></td>
<td>Saisissez le nombre de tâches Reduce. Pour plus d’informations concernant les fonctionnalités de parallélisation, consultez le manuel de référence de Pig Latin.</td>
</tr>
</tbody>
</table>

**Du côté de la sortie :**

<table>
<thead>
<tr>
<th>Propriétés du flux de sortie</th>
<th>Valeur</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Catch Output Reject</em></td>
<td>True ; False.</td>
</tr>
<tr>
<td><em>Catch Lookup Inner Join Reject</em></td>
<td>True ; False.</td>
</tr>
</tbody>
</table>
tPigMap

Propriétés du tPigMap Standard
Ces propriétés sont utilisées pour configurer le tPigMap s'exécutant dans le framework de Jobs
Standard.
Le composant tPigMap Standard appartient aux familles Big Data et Processing.
Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de
Talend.
Basic settings
Mapping links display as

Auto : par défaut, les liens sont en forme de courbes.
Curves : les liens du mapping sont en forme de courbes.
Lines : les liens du mapping sont en forme de lignes
droites. Cette dernière option améliore légèrement les
performances.

Map editor

Le Map Editor vous permet de définir les propriétés du
routage et des transformations du tPigMap.

Advanced settings
tStatCatcher
Statistics

Cochez la case afin de collecter les données de log au niveau de chaque composant.

Global Variables
Global Variables

ERROR_MESSAGE : message d'erreur généré par le
composant lorsqu'une erreur survient. Cette variable est
une variable After et retourne une chaîne de caractères.
Cette variable fonctionne uniquement si la case Die on
error est décochée, si le composant a cette option.
Une variable Flow fonctionne durant l'exécution
d'un composant. Une variable After fonctionne après
l'exécution d'un composant.
Pour renseigner un champ ou une expression à l'aide
d'une variable, appuyez sur les touches Ctrl+Espace
pour accéder à la liste des variables. A partir de cette
liste, vous pouvez choisir la variable que vous souhaitez
utiliser.
Pour plus d'informations concernant les variables,
consultez le Guide utilisateur du Studio Talend.

Utilisation
Règle d'utilisation

Plusieurs utilisations sont possibles, de la simple
réorganisation des champs de données aux
transformations les plus complexes, telles que le
multiplexage et le démultiplexage de données, la con
caténation, l'inversion, le filtrage, la division, etc., dans
un processus Pig.

3156


Limitation

L’utilisation du tPigMap requiert des connaissances minimales en Java et Pig Latin afin d’exploiter au mieux ses fonctionnalités.

Ce composant est une étape de jonction. Il ne peut donc être ni un composant d’entrée, ni un composant de sortie.

Scénario : Effectuer une jointure dans un processus Pig sur des données concernant les conditions de circulation

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Le Job de ce scénario utilise deux composants tPigLoad pour lire des données concernant les conditions de circulation et les événements associés sur les routes données depuis une distribution Hadoop, effectuer une jointure et filtrer les données à l’aide du tPigMap, puis d’écrire les résultats dans cette distribution Hadoop à l’aide de deux tPigStoreResult.

La distribution Hadoop à utiliser conserve les données concernant les conditions de circulations (normale ou embouteillage) ainsi que les données concernant les événements liés, tels que des travaux, la pluie ou aucun événement. Dans cet exemple, les données à utiliser se présentent comme suit :

1. Les données concernant les conditions de circulation sont stockées dans le répertoire /user/ychen/tpigmap/date&traffic. jam signifie qu’il y a un embouteillage, normal signifie que la circulation est normale.

2013-01-11 00:27:53;Bayshore Freeway;jam
2013-02-28 07:01:18;Carpinteria Avenue;jam
2013-01-26 11:27:59;Bayshore Freeway;normal
2013-03-07 20:48:51;South Highway;jam
2013-02-07 07:40:10;Lindbergh Blvd;normal
2013-01-22 17:13:55;Pacific Hwy S;normal
2013-03-17 23:12:26;Carpinteria Avenue;normal
2013-01-15 08:06:53;San Diego Freeway;jam
2. Les données relatives aux événements sont stockées dans le répertoire `/user/ychen/tpigmap/`.


date\&event
road work signifie qu’il y a des travaux, rain signifie qu’il pleut et no event signifie qu’il n’y a pas d’événement particulier.

Pour chaque heure affichée dans les données, une ligne concernant les conditions de trafic et une ligne concernant les événements sont enregistrées. Vous devez effectuer une jointure sur les données d’une table afin de détecter facilement l’impact des événements sur le trafic routier.

Pour reproduire le scénario, assurez-vous que le Studio Talend à utiliser possède les droits de lecture et d’écriture dans la distribution Hadoop puis procédez comme suit :

**Relier les composants**

**Procédure**

1. Dans la perspective Integration du Studio Talend, créez un Job vide nommé pigweather, par exemple, depuis le nœud Job Designs dans la vue Repository.

Pour plus d’informations concernant la création d’un Job, consultez le Guide utilisateur du Studio Talend.

2. Déposez deux composants tPigLoad, un tPigMap et deux tPigStoreResult de la Palette de l’espace de modélisation graphique.


3. Cliquez-droit sur le tPigLoad nommé traffic pour le connecter au tPigMap à l’aide d’un lien Row > Pig combine, dans le menu contextuel.

4. Répétez l’opération pour relier le tPigLoad nommé event au tPigMap également. Lorsque le deuxième lien est créé, il devient automatiquement un lien Lookup.

5. Utilisez le lien Row > Pig combine pour connecter le tPigMap à chaque composant tPigStoreResult.

Renommez ces liens dans la boîte de dialogue qui apparaît une fois que vous avez sélectionné le type de lien dans le menu contextuel. Dans ce scénario, nommez out le lien vers le tPigStoreResult nommé normal et reject le lien vers le composant tPigStoreResult nommé jam.
Configurer le tPigLoad

Charger les données relatives à la circulation

Procédure

1. Double-cliquez sur le tPigLoad nommé *traffic* pour ouvrir sa vue Component.

![Image of tPigLoad configuration](image)

2. Cliquez sur le bouton \(\ldots\) à côté du champ *Edit schema* pour ouvrir l'éditeur de schéma.

3. Cliquez trois fois sur le bouton \(\ldots\) pour ajouter trois lignes et, dans la colonne *Column*, renommez-les respectivement *date*, *street* et *traffic*.

![Image of schema editor](image)

4. Cliquez sur *OK* pour valider les modifications.
5. Dans la zone **Mode**, sélectionnez l’option **Map/Reduce**, puisque le **Studio Talend** doit se connecter à une distribution Hadoop distante.

6. Dans les listes **Distribution** et **Version**, sélectionnez la distribution Hadoop à utiliser. Dans cet exemple, sélectionnez **Hortonworks Data Platform V1.0.0**.

7. Dans la liste **Load function**, sélectionnez la fonction **PigStorage** pour lire les données source, puisque les données sont contenues dans un fichier structuré au format UTF-8, humainement lisible.


9. Dans le champ **Input file URI**, saisissez le chemin d’accès au répertoire dans lequel sont stockées les informations relatives à la circulation. Dans cet exemple, le chemin d’accès au répertoire est `/user/ychen/tpigmap/date&traffic`.

10. Dans le champ **Field separator**, saisissez `;` ou tout autre séparateur utilisé dans les données source.

**Charger les données relatives aux événements**

**Procédure**

1. Double-cliquez sur le **tPigLoad** nommé `event` pour ouvrir sa vue **Component**.

   ![Component tPigLoad](image)

   2. Cliquez sur le bouton ⚙ à côté du champ **Edit schema** pour ouvrir l’éditeur de schéma.

   3. Cliquez trois fois sur le bouton + pour ajouter trois lignes et, dans la colonne **Column**, renommez-les respectivement `date`, `street` et `event`.
4. Cliquez sur OK pour valider les modifications.

5. Dans la zone Mode, sélectionnez Map/Reduce.
   Puisque vous avez déjà configuré la connexion à la distribution Hadoop, dans le premier
   composant tPigLoad nommé traffic, le composant event réutilise cette connexion. Ainsi, la
   Distribution et la Version sont automatiquement sélectionnées.

6. Dans le champ Load function, sélectionnez la fonction PigStorage pour lire les données source.

7. Dans le champ Input file URI, saisissez le chemin d'accès au répertoire dans lequel les données
   relatives aux événements sont stockées. Dans cet exemple, le chemin d'accès au répertoire est "/
   user/ychen/tpigmap/date&event".

Configurer le tPigMap

1. Double-cliquez sur le composant tPigMap pour ouvrir sa vue Map Editor.

Créer le schéma de sortie

Procédure

   La table supérieure représente le flux d’entrée principal et la table inférieure représente le flux de
   référence (Lookup).
   Du côté de la sortie (droite), les deux tables représentent les flux de sortie précédemment nommés
   out1 et reject.
   De la table du flux principal, déposez les trois colonnes dans chaque table de sortie.

2. Du flux Lookup, déposez la colonne event dans chaque table de sortie.
   Dans la vue Schema editor, vous pouvez voir les schémas terminés des deux côtés sont terminés.
   Vous pouvez cliquez sur chaque table pour afficher la schéma dans cette vue.
Configurer les conditions de mapping

Procédure

1. Dans la table du flux de référence, cliquez sur le bouton pour ouvrir le panneau des paramètres.
2. Dans la ligne Join Model, sélectionnez Left Outer Join pour vous assurer que tous les enregistrements du flux principal sont inclus dans cette jointure.
3. Dans la table de sortie out1, cliquez sur le bouton pour afficher le champ d'expression de filtre.
4. Saisissez 'normal'== row1.traffic
   Cela permet au tPigMap d'écrire en sortie uniquement les enregistrements contenant le mot normal dans le flux out1.
5. Dans la table de sortie reject, cliquez sur le bouton pour ouvrir le panneau des paramètres.
6. Paramétrez l’option Catch Output Reject à true pour écrire en sortie les enregistrements contenant le mot jam dans le flux reject.
7. Cliquez sur Apply puis sur OK pour valider les modifications et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

Configurer le tPigStoreResult

Procédure

1. Double-cliquez sur le tPigStoreResult nommé normal pour ouvrir sa vue Component.

![Component](image)

2. Dans le champ Result file, saisissez le chemin d'accès au répertoire dans lequel écrire les résultats. Dans ce scénario, le chemin d'accès au répertoire est /user/ychen/tpigmap/traffic_normal. Ce répertoire reçoit les enregistrements normal.
3. Cochez la case Remove result directory if exists.
4. Dans la liste Store function, sélectionnez PigStorage pour écrire les enregistrements au format UTF-8 humainement lisible.
5. Dans le champ Field separator, saisissez :.
6. Répétez les opérations pour configurer le tPigStoreResult nommé jam, mais, dans le champ Result file, saisissez le chemin d'accès au répertoire /user/ychen/tpigmap/traffic_jam.
Résultats

Remarque :
Si l’un des composants ne récupère pas son schéma via le tPigMap, une icône d’avertissement apparaît. Dans ce cas, cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent. L’icône d’avertissement disparaît.

Exécuter le Job

Appuyez sur F6 pour exécuter le Job.
Cela fait, vérifiez les résultats dans la distribution Hadoop utilisée.

File: /user/ychen/tpigmap/traffic_jam/part-r-00000

Dans les enregistrements traffic_jam, vous pouvez regarder quel événement correspond à un embouteillage et dans les enregistrements traffic_normal, vous pouvez regarder dans quelles situations la circulation est normale.

File: /user/ychen/tpigmap/traffic_normal/part-r-00000

Si vous souhaitez plus d’informations concernant le Job et son exécution, utilisez la console Web fournie par la distribution Hadoop que vous utilisez.
Dans le JobHistory, vous pouvez facilement trouver le statut d'exécution de votre Job Pig car il est automatiquement nommé. Son nom est le résultat de la concaténation du nom du projet contenant le Job, du nom du Job, de sa version et du libellé du premier composant tPigLoad utilisé. La convention de nommage d’un Job Pig dans le JobHistory est la suivante : ProjectName_JobNameVersion_FirstComponentName.
**tPigReplicate**

Ce composant effectue différentes opérations sur le même schéma.

Le tPigReplicate s’utilise à la suite d’un composant Pig d’entrée. Il duplique le schéma entrant en autant de flux de sortie identiques que nécessaire.

**Propriétés du tPigReplicate Standard**

Ces propriétés sont utilisées pour configurer le tPigReplicate s’exécutant dans le framework de Jobs Standard.

Le composant tPigReplicate Standard appartient aux familles Big Data et Processing.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
  • View schema : sélectionnez cette option afin de voir le schéma.
  • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
  • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].  
Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent. |
| Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
### Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l'exécution d’un composant. Une variable *After* fonctionne après l'exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

**Règle d’utilisation**

Ce composant n’est pas un composant de début (fond vert), il nécessite un **tPigLoad** en entrée et attend d’autres composants Pig pour gérer son (ses) flux de sortie.

**Prérequis**

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le **Studio Talend**. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le **Studio Talend** et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées `MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native`. Par exemple, pour Windows, la bibliothèque est `lib\MapRClient.dll` dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

  Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : `no MapRClient in java.library.path`.

- Configurez l’argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue [Preferences] dans le menu **Window**. Cet argument
fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

### Connections

Liens de sortie (de ce composant à un autre) :
- **Row** : Pig combine. Ce lien effectue une jointure sur tous les processus de données créées dans le Job et les exécute simultanément.

Lien d’entrée (d’un autre composant à celui-ci) :
- **Row** : Pig combine.

Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.

### Scénario : Répliquer un flux et trier deux flux identiques

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Le Job de ce scénario utilise des composants Pig pour gérer les noms et les états chargés d’un système HDFS donné. Il lit et réplique le flux d’entrée et trie les données des deux flux identiques, respectivement sur le nom et sur l’état, puis écrit les résultats dans ce système HDFS.

Avant de commencer à reproduire ce Job, assurez-vous de posséder les droits en lecture écriture de données dans la distribution Hadoop à utiliser et que Pig est correctement installé dans cette distribution.
Relier les composants

Procédure

1. Dans la perspective **Integration** du *Studio Talend*, créez un Job vide, nommé *Replicate* par exemple, à partir du nœud *Job Designs* dans la vue *Repository*.

   Pour plus d’informations concernant la création d’un Job, consultez le *Guide utilisateur du Studio Talend*.

2. Déposez un *tPigLoad*, un *tPigReplicate*, deux *tPigSort* et deux *tPigStoreResult* dans l’espace de modélisation graphique.

   Le composant *tPigLoad* lit des données du système HDFS donné. Les données d’exemple utilisées dans ce scénario se présentent comme suit :

   Andrew Kennedy;Mississippi
   Benjamin Carter;Louisiana
   Benjamin Monroe;West Virginia
   Bill Harrison;Tennessee
   Calvin Grant;Virginia
   Chester Harrison;Rhode Island
   Chester Hoover;Kansas
   Chester Kennedy;Maryland
   Chester Polk;Indiana
   Dwight Nixon;Nevada
   Dwight Roosevelt;Mississippi
   Franklin Grant;Nebraska

   L’emplacement des données dans ce scénario est le suivant */user/ychen/raw/Name&State.csv*.

3. Reliez les composants à l’aide de liens *Row > Pig combine*.

Configurer le *tPigLoad*

Procédure

1. Double-cliquez sur le *tPigLoad* pour ouvrir sa vue *Component*. 
2. Cliquez sur le bouton ... à côté du champ **Edit schema** pour ouvrir l’éditeur de schéma.

![Schema editor]

3. Cliquez deux fois sur le bouton + pour ajouter deux lignes et nommez-les respectivement **Name** et **State**.

4. Cliquez sur **OK** pour valider ces modifications et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

5. Dans la zone **Mode**, sélectionnez **Map/Reduce**. Le Hadoop à utiliser dans ce scénario est installé sur une machine distante. Une fois l’option sélectionnée, les paramètres à configurer apparaissent.

7. Dans la liste **Load function**, sélectionnez **PigStorage**.

8. Dans les champs **NameNode URI** et **ResourceManager**, saisissez l'emplacement du **NameNode** et du **ResourceManager** à utiliser pour Map/Reduce. Si vous utilisez WebHDFS, l'emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL, le schéma d'URI doit être `swebhdfs` et vous devez utiliser un `tLibraryLoad` dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.

9. Dans le champ **Input file URI**, saisissez l'emplacement des données à lire depuis HDFS. Dans cet exemple, l'emplacement est `/user/ychen/raw/NameState.csv`.

10. Dans le champ **Field separator**, saisissez `;`.

**Configurer le tPigReplicate**

**Procédure**

1. Double-cliquez sur le **tPigReplicate** pour ouvrir sa vue **Component**.

---

2. Cliquez sur le bouton à côté du champ **Edit schema** pour ouvrir l'éditeur de schéma et vérifier que le schéma est identique à celui du composant précédent.

---

**Remarque :**

Si le composant n'a pas le même schéma que celui du composant précédent, une icône d'avertissement apparaît. Dans ce cas, cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Cela fait, l'icône disparaît.
Configurer le tPigSort

Pourquoi et quand exécuter cette tâche

Deux composants tPigSort sont utilisés pour trier les deux flux de sortie identiques : un sur la colonne Name et l’autre sur la colonne State.

Procédure

1. Double-cliquez sur le premier composant tPigSort pour ouvrir sa vue Component et définir le tri par nom.

2. Dans la table Sort key, ajoutez une ligne en cliquant sur le bouton + sous la table.

3. Dans la colonne Column, sélectionnez Name dans la liste déroulante et sélectionnez ASC dans la colonne Order.

4. Double-cliquez sur l’autre composant tPigSort pour ouvrir sa vue Component et définir le tri par état.

5. Dans la table Sort key, ajoutez une ligne puis sélectionnez Name dans la liste déroulante de la colonne Column et sélectionnez ASC dans la colonne Order.

Configurer le tPigStoreResult

Pourquoi et quand exécuter cette tâche

Deux composants tPigStoreResult sont utilisés pour écrire les données triées dans HDFS.
**Procédure**

1. Double-cliquez sur le premier composant **tPigStoreResult** pour ouvrir sa vue **Component** et écrire les données triées par nom.

2. Dans le champ **Result file**, saisissez le répertoire dans lequel les données seront écrites. Ce répertoire sera créé s’il n’existe pas. Dans ce scénario, le répertoire est le suivant `/user/ychen/sort/tPigreplicate/byName.csv`.

3. Sélectionnez **Remove result directory if exists**.

4. Dans la liste **Store function**, sélectionnez **PigStorage**.

5. Dans le champ **Field separator**, saisissez `;`.

6. Répétez l’opération pour l’autre composant **tPigStoreResult** mais sélectionnez un répertoire différent pour les données triées par état. Dans ce scénario, le répertoire est le suivant `/user/ychen/sort/tPigreplicate/byState.csv`.

**Exécuter le Job**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez exécuter le Job.

**Procédure**

Appuyez sur **F6** pour l’exécuter.

**Résultats**

Cela fait, parcourez votre système jusqu’aux emplacements où les résultats ont été écrits dans HDFS.

La capture d’écran suivante présente les résultats triés par nom :

![Image du composant tPigStoreResult](image-url)
La capture d'écran suivante présente les résultats triés par état :

File: /user/ychen/sort/tPigreplicate/byState.csv/part-r-00000

Si vous souhaitez plus d'informations concernant le Job et son exécution, utilisez la console Web fournie par la distribution Hadoop que vous utilisez.
Dans le JobHistory, vous pouvez facilement trouver le statut d'exécution de votre Job Pig car il est automatiquement nommé. Son nom est le résultat de la concaténation du nom du projet contenant le Job, du nom du Job, de sa version et du libellé du premier composant tPigLoad utilisé. La convention de nommage d'un Job Pig dans le JobHistory est la suivante : ProjectName_JobNameVersion_FirstComponentName.
tPigSort

Ce composant est utilisé pour trier une relation à partir d’une ou plusieurs clé(s) défini(s).

**Propriétés du tPigSort Standard**

Ces propriétés sont utilisées pour configurer le tPigSort s’exécutant dans le framework de Jobs Standard.

Le composant tPigSort Standard appartient aux familles Big Data et Processing.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

### Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort key</td>
<td>Cliquez sur le bouton <strong>Add</strong> sous la table <strong>Sort key</strong> pour ajouter une ou plusieurs ligne(s) afin de spécifier une colonne et trier selon chaque clé de tri.</td>
</tr>
</tbody>
</table>
### Advanced settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase parallelism</td>
<td>Cochez cette case pour définir le nombre de tâches &quot;reduce&quot; pour des Jobs MapReduce.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ce composant est généralement utilisé comme étape intermédiaire. Il nécessite donc un composant d’entrée et un composant de sortie.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prérequis</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.</td>
</tr>
<tr>
<td></td>
<td>- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : <a href="http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr">http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr</a> (en anglais). Si vous n’ajoutez pas de librairie, il est possible que vous rencontreriez l’erreur suivante : no MapRClient in java.library.path.</td>
</tr>
</tbody>
</table>
• Configurez l'argument `-Djava.library.path`, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d'accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d'utiliser entièrement l'aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d'informations concernant l'installation d'une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

| Limitation | La connaissance des scripts Pig est nécessaire. |

Scénario : Trier des données en ordre ascendant

Ce scénario s'applique uniquement aux solutions Talend avec Big Data.

Ce scénario décrit un Job à trois composants triant des lignes de données à partir d'une ou plusieurs condition(s) de tri et stockant le résultat dans un fichier local.

Construire le Job

Procédure

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : tPigSort, tPigLoad, tPigStoreResult.
2. Connectez le tPigLoad au tPigFilterRow à l'aide d'un lien Row > Pig Combine.
3. Connectez le tPigFilterRow au tPigStoreResult à l'aide d'un lien Row > Pig Combine.

Charger les données

Procédure

1. Double-cliquez sur le composant tPigLoad pour ouvrir sa vue Basic settings.
2. Cliquez sur le bouton [...] à côté du champ **Edit schema** afin d’ajouter des colonnes au schéma du composant **tPigLoad**.

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Type</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>✔</td>
<td>String</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td></td>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td></td>
<td>Character</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Cliquez sur le bouton **Add** afin d’ajouter les colonnes *Name*, *Country* et *Age* et cliquez sur **OK** pour sauvegarder la configuration.

4. Sélectionnez **Local** dans la liste **Mode**.

5. Renseignez le champ **Input filename** avec le chemin d’accès au fichier d’entrée.
   Dans ce scénario, le fichier d’entrée est *CustomerList*. Il contient les lignes de noms, les noms des pays et les âges des clients.

6. Sélectionnez **PigStorage** dans la liste **Load function**.

7. Laissez les autres paramètres tels qu’ils sont.

**Paramétrer la condition de tri**

**Procédure**

1. Double-cliquez sur le **tPigSort** afin d’ouvrir sa vue **Basic settings**.
2. Cliquez sur **Sync columns** pour récupérer la structure du schéma du composant précédent.

3. Cliquez sur le bouton **Add** sous la table **Sort key** afin d'ajouter une nouvelle clé de tri. Sélectionnez **Age** dans la liste **Column** et sélectionnez **ASC** dans la liste **Order**.

Cette clé de tri va trier les données dans la liste **CustomerList** en ordre ascendant à partir de l’**Age**.

### Sauvegarder les données dans un fichier local

**Procédure**

1. Double-cliquez sur le composant **tPigStoreResult** afin d'ouvrir sa vue **Basic settings**.

2. Cliquez sur **Sync columns** afin de récupérer la structure du schéma du composant précédent.

3. Cochez la case **Remove result directory if exists**.

4. Renseignez le champ **Result file** avec le chemin d'accès au fichier de résultat.

   Dans ce scénario, le résultat du filtre est sauvegardé dans le fichier **Lucky_Customer**.

5. Sélectionnez **PigStorage** dans la liste **Store function**.

6. Laissez les autres paramètres tels qu’ils sont.

### Exécuter le Job

Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.
Le fichier *Lucky_Customer* est généré et contient les données en ordre ascendant selon la colonne *Age*. 

<table>
<thead>
<tr>
<th>Name</th>
<th>Country</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Romeo</td>
<td>UK</td>
<td>19</td>
</tr>
<tr>
<td>Silvia</td>
<td>Spain</td>
<td>20</td>
</tr>
<tr>
<td>Billy</td>
<td>Canada</td>
<td>21</td>
</tr>
<tr>
<td>Mike</td>
<td>USA</td>
<td>22</td>
</tr>
<tr>
<td>Manik</td>
<td>Marta</td>
<td>23</td>
</tr>
<tr>
<td>Juan</td>
<td>Cuba</td>
<td>23</td>
</tr>
<tr>
<td>Bill</td>
<td>USA</td>
<td>24</td>
</tr>
<tr>
<td>Natasha</td>
<td>Russia</td>
<td>25</td>
</tr>
<tr>
<td>Huamel</td>
<td>China</td>
<td>26</td>
</tr>
<tr>
<td>Ahmad</td>
<td>KSA</td>
<td>27</td>
</tr>
<tr>
<td>Fenricka</td>
<td>Canada</td>
<td>27</td>
</tr>
<tr>
<td>Selena</td>
<td>Mexico</td>
<td>27</td>
</tr>
<tr>
<td>Pantalon</td>
<td>Peru</td>
<td>28</td>
</tr>
<tr>
<td>Didi</td>
<td>Brazil</td>
<td>28</td>
</tr>
<tr>
<td>Julio</td>
<td>Peru</td>
<td>29</td>
</tr>
<tr>
<td>Zidane</td>
<td>Algeria</td>
<td>29</td>
</tr>
<tr>
<td>Nancy</td>
<td>Lebanon</td>
<td>31</td>
</tr>
<tr>
<td>Simao</td>
<td>Brazil</td>
<td>33</td>
</tr>
<tr>
<td>Eminem</td>
<td>USA</td>
<td>35</td>
</tr>
<tr>
<td>Ricky</td>
<td>Puerto Rico</td>
<td>37</td>
</tr>
<tr>
<td>Mario</td>
<td>Puerto Rico</td>
<td>49</td>
</tr>
<tr>
<td>Bob</td>
<td>Jamaica</td>
<td>55</td>
</tr>
<tr>
<td>Gaddafi</td>
<td>Libya</td>
<td>69</td>
</tr>
<tr>
<td>Toyota</td>
<td>Japan</td>
<td>75</td>
</tr>
<tr>
<td>Name</td>
<td>Country</td>
<td>Age</td>
</tr>
</tbody>
</table>
tPigStoreResult

Ce composant stocker les résultats d’un Job Pig dans un espace de stockage de données.

Propriétés du tPigStoreResult Standard

Ces propriétés sont utilisées pour configurer le tPigStoreResult s’exécutant dans le framework de Jobs Standard.

Le composant tPigStoreResult Standard appartient aux familles Big Data et Processing.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Property type | Peut être Repository ou Built-in.  
|              | L’option Repository vous permet de réutiliser les propriétés de connexion stockées dans le nœud Hadoop cluster du Repository. Une fois sélectionnée, le bouton apparaît. Cliquez dessus afin d’afficher la liste des propriétés stockées et, à partir de cette liste, sélectionnez les propriétés que vous souhaitez utiliser. Les paramètres appropriés sont alors définis automatiquement.  
|              | Sinon, si vous sélectionnez l’option Built-in, vous devez paramétrer manuellement chacun des paramètres.  

Schema et Edit Schema

| Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
| Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
| • View schema : sélectionnez cette option afin de voir le schéma.  
| • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
<p>| • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |</p>
<table>
<thead>
<tr>
<th><strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</th>
</tr>
</thead>
</table>
| **Use S3 endpoint** Cochez cette case pour écrire des données dans un dossier d’un bucket Amazon S3. Après avoir coché la case **Use S3 endpoint**, vous devez renseigner les paramètres suivants dans les champs qui apparaissent :
  - **S3 bucket name and folder** : saisissez le nom du bucket et celui du dossier dans lequel vous souhaitez écrire les données. Vous devez séparer le nom du bucket et le nom du dossier par une barre oblique (/).
  - **Access key et Secret key** : renseignez les informations d’authentification requises pour vous connecter au bucket Amazon S3 à utiliser.
    
    Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.
    
    Notez que le format du fichier S3 est S3N (S3 Native Filesystem).
| **Result folder URI** Spécifiez le chemin d’accès au fichier de résultat dans lequel les données sont stockées. |
| **Remove result directory if exists** Cochez cette case pour supprimer un répertoire de résultats existant. |
| **Remarque** :
Cette case est désactivée lorsque vous sélectionnez **HCatStorer** dans la liste **Store function**. |
| **Store function** Sélectionnez une fonction de stockage pour stocker les données :
  - **PigStorage** : Stocke les données au format UTF-8.
  - **BinStorage** : Stocke les données dans un format lisible par les machines.
  - **PigDump** : Stocke les données en tant que tuples dans un format UTF-8 lisible par l’humain.
  - **HCatStorer** : Stocke les données dans les tables gérées par HCatalog à l'aide de scripts Pig.
  - **HBaseStorage** : Stocke les données dans HBase. Vous devez terminer la configuration de HBase dans la zone **HBase configuration**.
  - **SequenceFileStorage** : Stocke des données au format SequenceFile. Vous devez terminer la configuration |
du fichier à stocker dans la zone **Sequence Storage Configuration** qui s’affiche.

- **RCFilePigStorage** : Stocke des données au format RCFile format.

- **AvroStorage** : Charge des fichiers Avro. Pour plus d’informations concernant AvroStorage, consultez la documentation Apache sur le site [https://cwiki.apache.org/confluence/display/PIG/AvroStorage](https://cwiki.apache.org/confluence/display/PIG/AvroStorage) (en anglais).

- **ParquetStorer** : Stocke des fichiers Parquet. Dans la liste **Associate tPigLoad component**, sélectionnez le composant **tPigLoad** dans lequel la connexion au cluster MapReduce à utiliser est définie.

Dans la liste **Compression** qui apparaît, sélectionnez le mode de compression à utiliser pour gérer le fichier Parquet. Le mode par défaut est **Uncompressed**.

- **Custom** : Stocke les données à l’aide d’une fonction de stockage personnalisée. Pour ce faire, vous devez enregistrer, dans l’onglet **Advanced settings**, le fichier Jar contenant la fonction à utiliser, puis, dans le champ affiché à côté du champ **Store function**, spécifier cette fonction.

Notez que, lorsque le format de fichier à utiliser est **PARQUET**, il est possible qu’il vous soit demandé de trouver le fichier Jar Parquet spécifique et l’installer dans le studio.

- Lorsque le mode de connexion à Hive est **Embedded**, le Job est exécuté sur votre machine locale et appelle ce Jar installé dans le studio.

- Lorsque le mode de connexion à Hive est **Standalone**, le Job est exécuté sur le serveur hébergeant Hive et ce fichier Jar est envoyé au système HDFS du cluster auquel vous vous connectez. Assurez-vous d’avoir correctement défini l’URI du NameNode dans le champ correspondant de la vue **Basic settings**.

Ce fichier Jar est téléchargeable depuis le site Web d’Apache. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center ([https://help.talend.com](https://help.talend.com)).

<table>
<thead>
<tr>
<th>HCatalog Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Renseignez les champs suivants pour configurer les tables gérées de HCatalog dans HDFS (système de fichiers distribué Hadoop) :</strong></td>
</tr>
</tbody>
</table>

**Distribution et Version** :

Sélectionnez la distribution Hadoop correspondant à celle que vous avez définie dans la configuration de connexion du composant **tPigLoad**. Ce composant est utilisé dans le même process Pig que le **tPigStoreResult** courant.

Si le composant **tPigLoad** se connecte à une distribution Hadoop personnalisée, vous devez également
sélectionner Custom pour ce composant tPigStoreResult. Ensuite, la table Custom jar apparaît. Dans cette table, vous devez ajouter uniquement les fichiers jar requis pour la fonction Store function sélectionnée.

HCat metastore : Saisissez l’emplacement du metastore de HCatalog, qui est également le metastore de Hive.

Database : Base de données dans laquelle les tables sont stockées.

Table : Table dans laquelle les données sont stockées.

Partition filter : Renseignez ce champ avec les clés de partitions afin de lister les partitions par filtre.

Remarque :

### HBase configuration

Cette zone est disponible pour la fonction HBaseStorage. Les paramètres à configurer sont :

**Distribution et Version :**

Sélectionnez la distribution Hadoop correspondant à celle que vous avez définie dans la configuration de connexion du composant tPigLoad. Ce composant est utilisé dans le même process Pig que le tPigStoreResult courant.

Si le composant tPigLoad se connecte à une distribution Hadoop personnalisée, vous devez également sélectionner Custom pour ce composant tPigStoreResult. Ensuite, la table Custom jar apparaît. Dans cette table, vous devez ajouter uniquement les fichiers Jar requis pour la fonction Store function sélectionnée.

**Zookeeper quorum :**

Saisissez le nom ou l’URL du service Zookeeper utilisé pour coordonner les transactions entre votre Studio et votre base de données. Notez que, lorsque vous configurez Zookeeper, vous pouvez avoir besoin de configurer explicitement la propriété zookeeper.znode.parent pour définir le chemin vers le nœud znode racine contenant tous les znodes créés et utilisés par votre base de données. Cochez la case la case Set Zookeeper znode parent afin de définir cette propriété.

**Zookeeper client port :**

Saisissez le numéro du port d’écoute client du service Zookeeper que vous utilisez.

**Table name :**
Saisissez le nom de la table HBase dans laquelle vous souhaitez stocker des données. La table doit exister dans la cible HBase.

**Row key column** :
Sélectionnez la colonne utilisée comme colonne Row Key de la table HBase.

**Store row key column to Hbase column** :
Cochez cette case pour que la colonne Row Key soit une colonne HBase appartenant à une famille de colonnes spécifique.

**Mapping** :
Renseignez cette table afin de mapper les colonnes de la table à utiliser avec les colonnes du schéma défini pour le flux de données à traiter.

La colonne **Column** de cette table est automatiquement renseignée une fois le schéma défini. Dans la colonne **Family name**, saisissez les familles de colonnes que vous souhaitez créer ou utiliser pour grouper les colonnes dans la colonne **Column**. Pour plus d'informations concernant une famille de colonnes, consultez la documentation Apache à l'adresse suivante : Column families (en anglais).

**Field separator**
Saisissez un caractère, une chaîne de caractères ou une expression régulière pour séparer les champs des données transférées.

*Remarque :*
Ce champ est activé uniquement lorsque vous sélectionnez PigStorage dans la liste Store function.

**Sequence Storage configuration**
Cette zone est disponible uniquement pour la fonction SequenceFileStorage. Comme un enregistrement SequenceFile se compose de paires binaires clé/valeur, les paramètres à configurer sont les suivants :

**Key column** : Sélectionnez la colonne clé d'un enregistrement clé/valeur.

**Value column** : Sélectionnez la colonne de valeur d’un enregistrement clé/valeur.

**Advanced settings**

**Register jar**
Cliquez sur le bouton [+] pour ajouter des lignes à la table et, dans ces lignes, parcourez votre système jusqu’aux fichiers Jar à ajouter. Par exemple, pour enregistrer un fichier Jar nommé piggybank.jar, cliquez sur le bouton [+] pour ajouter une ligne, puis cliquez sur cette ligne pour afficher le bouton [...] (permettant de parcourir votre système) puis cliquez sur ce bouton pour parcourir votre système jusqu’au fichier piggybank.jar, dans l’assistant [Select Module] qui s’ouvre.
### HBaseStorage configuration

Ajoutez et configurez plus d'options de stockage HBaseStorage dans cette table. Les options sont :

- **loadKey** : saisissez `true` pour stocker la colonne Row Key comme première colonne du schéma de résultat. Sinon, saisissez `false`,
- **gt** : la valeur minimale de clé,
- **lt** : la valeur maximale de clé,
- **gte** : la valeur minimale de clé (comprise),
- **lte** : la valeur maximale de clé (comprise),
- **limit** : nombre maximum de lignes à récupérer par région,
- **caching** : nombre de lignes à mettre en cache,
- **caster** : le convertisseur à utiliser pour écrire des valeurs dans HBase, par exemple `Utf8StorageConverter`.

### Define the jars to register for HCatalog

Cette case apparaît lorsque vous utilisez `tHCatStorer`, mais vous pouvez la laisser décochée, car le Studio enregistre automatiquement les fichiers Jar requis. Si le fichier Jar est manquant, vous pouvez cocher cette case pour afficher la table `Register jar for HCatalog` et configurer le chemin d'accès à ce Jar manquant.

### tStatCatcher Statistics

Cochez cette case pour收集 les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case `Die on error` est décochée, si le composant a cette option.

Une variable `Flow` fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**

Ce composant est utilisé pour terminer une séquence Pig et nécessite un `tPigLoad` au début de cette séquence afin de fournir les données.
<table>
<thead>
<tr>
<th>Prérequis</th>
</tr>
</thead>
<tbody>
<tr>
<td>La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.</td>
</tr>
<tr>
<td>- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : <a href="http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr">http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr</a> (en anglais). Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.</td>
</tr>
<tr>
<td>- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR. Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>La connaissance des scripts Pig est nécessaire. Si vous sélectionnez HCatStorer dans la liste Store function, la connaissance de HCatalog DDL (HCatalog Data Definition Language, un sous-ensemble de Hive Data Definition Language) est nécessaire. Pour plus d’informations concernant HCatalog DDL, consultez <a href="https://cwiki.apache.org/confluence/display/Hive/HCatalog">https://cwiki.apache.org/confluence/display/Hive/HCatalog</a> (en anglais).</td>
</tr>
</tbody>
</table>
Scénario associé

- Pour un scénario associé dans lequel le composant `tPigStoreResult` utilise le mode `Local`, consultez Scénario : Trier des données en ordre ascendant à la page 3177 du `tPigSort`.

- Pour un scénario associé dans lequel le `tPigStoreResult` utilise le mode `Map/Reduce`, consultez Scénario : Charger une table HBase à la page 3150.
tPivotToColumnsDelimited

Ce composant permet d’ajuster la sélection de données à transmettre en sortie.
Le tPivotToColumnsDelimited extrait les données en fonction d’une opération d’agrégation exercée sur une colonne pivot.

Propriétés du tPivotToColumnsDelimited Standard

Ces propriétés sont utilisées pour configurer le tPivotToColumnsDelimited s’exécutant dans le framework de Jobs Standard.
Le composant tPivotToColumnsDelimited Standard appartient à la famille File.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pivot column</td>
<td>Sélectionnez la colonne qui sert de pivot pour l’opération d’agrégation dans le flux d’entrée.</td>
</tr>
<tr>
<td>Aggregation column</td>
<td>Sélectionnez la colonne contenant les données à agréger dans le flux d’entrée.</td>
</tr>
<tr>
<td>Aggregation function</td>
<td>Sélectionnez la fonction à utiliser dans le cas où plusieurs valeurs sont disponibles dans la colonne pivot.</td>
</tr>
<tr>
<td>Group by</td>
<td>Définissez les ensembles d’agrégation dont les valeurs qui sont utilisées pour les calculs.</td>
</tr>
</tbody>
</table>

**Input Column**: Faites le lien entre les libellés des colonnes d’entrée et ceux des colonnes de sortie, dans le cas où vous souhaitez que les libellés du schéma de sortie soient différents du schéma d’entrée.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field separator</td>
<td>Caractère, chaîne ou expression régulière pour séparer les champs du fichier de sortie.</td>
</tr>
<tr>
<td>Row separator</td>
<td>Chaîne (ex : ‘\n’ sous Unix) séparant les lignes.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_OUT</td>
<td>Nombre de lignes écrites dans le fichier par le composant. Cette variable est une variable After et retourne un entier.</td>
</tr>
</tbody>
</table>
**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant nécessite un flux d’entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Limitation</strong></td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

**Scénario : Utiliser une colonne pivot pour agréger des données**

Le scénario suivant décrit un Job qui agrège des données d’un fichier d’entrée délimité à l’aide d’une colonne pivot.

**Déposer et relier les composants**

**Procédure**

2. Reliez les composants à l’aide d’un lien Row > Main.
Configurer les composants

Paramétrer le composant d’entrée

Procédure

1. Double-cliquez sur le tFileInputDelimited pour afficher sa vue Basic settings.

   Property Type: Built-In
   File name/Stream: "D:\Input\pivot\n.csv"
   Row Separator: "\n"  Field Separator: ;
   CSV options: Header, Footer, Limit
   Schema: Built-In

2. Parcourez votre système jusqu’au fichier d’entrée dans le champ File Name.
   Le fichier à utiliser se compose de trois colonnes, ID, Question et Answer.

<table>
<thead>
<tr>
<th>Id</th>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Name</td>
<td>Juan</td>
</tr>
<tr>
<td>2</td>
<td>Name</td>
<td>Jean</td>
</tr>
<tr>
<td>3</td>
<td>Name</td>
<td>John</td>
</tr>
<tr>
<td>1</td>
<td>Gender</td>
<td>M</td>
</tr>
<tr>
<td>2</td>
<td>Gender</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>Gender</td>
<td>M</td>
</tr>
<tr>
<td>1</td>
<td>Surgery</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Surgery</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>Surgery</td>
<td>Yes</td>
</tr>
<tr>
<td>1</td>
<td>Age</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>Age</td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td>Age</td>
<td>42</td>
</tr>
<tr>
<td>1</td>
<td>Name</td>
<td>Mary</td>
</tr>
</tbody>
</table>

3. Définissez les séparateurs de lignes (Row) et de champs (Field), respectivement : retour chariot et point virgule, dans cet exemple.

4. Le fichier contient un en-tête, ainsi définissez-le.

5. Paramétrez le schéma avec les trois colonnes : ID, Questions, Answers.

Paramétrer le composant de sortie

Procédure

1. Puis sélectionnez le composant tPivotToColumnsDelimited et paramétrez ses propriétés dans l’onglet Basic settings de la vue Component.
2. Dans le champ **Pivot column**, sélectionnez la colonne pivot dans le schéma d’entrée. La colonne pivot est généralement celle présentant le plus de doublons (valeurs d’agrégation pivot).

3. Dans le champ **Aggregation column**, sélectionnez la colonne à agréger dans le schéma d’entrée.

4. Dans le champ **Aggregation function**, sélectionnez la fonction à effectuer lorsque des doublons sont trouvés.

5. Dans la table **Group by**, ajoutez une entrée dans la colonne **Input column**, qui sera utilisée pour regrouper la colonne agrégée.

6. Dans le champ **File Name**, indiquez le chemin d’accès et le nom du fichier de sortie. Et dans les champs **Row** et **Field separator**, paramétrez les séparateurs pour les lignes et données de sortie ayant été agrégées.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour exécuter le Job.

**Résultats**

<table>
<thead>
<tr>
<th>id</th>
<th>Name</th>
<th>Gender</th>
<th>Surgery</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mary</td>
<td>M</td>
<td>Yes</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>Jean</td>
<td>M</td>
<td>No</td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td>John</td>
<td>F</td>
<td>Yes</td>
<td>42</td>
</tr>
</tbody>
</table>

Le fichier de sortie affiche les données qui viennent d’être agrégées.
tPOP

Ce composant récupère un ou plusieurs e-mails à partir d’un protocole POP3 ou IMAP.

Le tPOP utilise le protocole POP ou IMAP pour se connecter au serveur de messagerie spécifié. Il récupère ensuite un ou plusieurs e-mails et écrit les informations récupérées dans un ou plusieurs fichiers définis. Les paramètres avancés (Advanced settings) permettent d’appliquer un ou plusieurs filtres à la sélection.

Propriétés du tPOP Standard

Ces propriétés sont utilisées pour configurer le tPOP s’exécutant dans le framework de Jobs Standard.

Le composant tPOP Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

| Host | Renseignez le nom de l’hôte ou l’adresse IP du serveur de messagerie auquel vous voulez vous connecter. |
| Port | Renseignez le numéro du port du serveur de messagerie. |
| Username et Password | Renseignez les informations d’authentification à votre compte de messagerie :
Username : entrez l’identifiant que vous utilisez pour accéder à votre boîte mail.
Password : entrez le mot de passe de connexion à votre boîte mail. |
| Output directory | Renseignez le chemin d’accès au dossier dans lequel vous voulez stocker les e-mails récupérés à partir de votre messagerie, ou cliquez sur […] pour parcourir vos dossiers jusqu’à l’emplacement souhaité. |
| Filename pattern | Appuyez sur Ctrl+Espace pour afficher la liste des modèles prédéfinis ou définissez manuellement la syntaxe du nom des fichiers cible. Les données des e-mails récupérés à partir du serveur de messagerie sont écrites dans ces fichiers. |
| Retrieve all emails? | Cochez cette case pour récupérer tous les messages présents sur le serveur de messagerie. |
| Number of emails to retrieve | Saisissez le nombre de messages à récupérer. Ce champ est disponible uniquement lorsque la case Retrieve all emails? est décochée. |
| Newer email first | Cochez cette case pour récupérer les messages les plus récents, en respectant le nombre spécifié dans le champ Number of emails to retrieve. Les messages récupérés seront retournés par ordre chronologique. |
Cette case est disponible uniquement lorsque la case **Retrieve all emails?** (cochée par défaut) est décochée.

| **Delete emails from server** | Cochez cette case si vous ne souhaitez pas conserver sur le serveur les e-mails récupérés.  

**Remarque :** 
Pour les serveurs Gmail, cette option ne fonctionne pas avec le protocole POP3. Sélectionnez le protocole IMAP et vérifiez que le compte Gmail est configuré pour utiliser ce protocole. |
|---|---|

| **Choose the protocol** | Sélectionnez à partir de la liste déroulante le protocole à utiliser pour récupérer les e-mails. Le protocole dépend de celui utilisé par votre messagerie. Si vous choisissez le protocole imap, vous aurez la possibilité de spécifier le dossier contenant les e-mails que vous souhaitez récupérer. |

| **Use SSL** | Cochez cette case si votre serveur de messagerie utilise ce protocole de sécurisation des échanges.  

**Remarque :** 
Cette option est obligatoire pour les utilisateurs de messagerie Gmail. |
|---|---|

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>tStatCatcher Statistics</strong></th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

| **Filter** | Cliquez sur le bouton [+] pour ajouter des lignes de filtre afin de ne récupérer qu’une sélection d’e-mails : |

**Filter item** : sélectionnez l’un des types de filtre suivants à partir de la liste déroulante :  
*From* : les messages sont filtrés en fonction de l’adresse e-mail de provenance.  
*To* : les messages sont filtrés en fonction de l’adresse e-mail de destination.  
*Subject* : les messages sont filtrés en fonction du sujet.  
*Before date* : les messages sont filtrés en fonction de la date d’émission ou de réception ; ils doivent être antérieurs à la date indiquée.  
*After date* : les messages sont filtrés en fonction de la date d’émission ou de réception ; ils doivent être postérieurs à la date indiquée.  

**Pattern** : appuyez sur Ctrl+Espace pour afficher la liste des valeurs disponibles et sélectionnez la valeur à utiliser pour chacun des filtres. |

| **Filter condition relation** | Sélectionnez le type de relation à appliquer entre les filtres : |
and : les conditions déterminées par les filtres s’ajoutent les unes aux autres, la recherche est plus restrictive.
or : les conditions déterminées par les filtres sont indépendantes les unes des autres, la recherche est plus large.

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant ne traite pas de flux de données, il peut donc être utilisé seul. |
| Limitation | Lorsque la case Use SSL est cochée ou que le protocole imap est sélectionné, le tPOP ne fonctionne pas avec IBM Java 6. |

**Scénario : Récupérer une sélection d’e-mails à partir d’un compte de messagerie**

Ce scénario à un seul composant a pour but de récupérer un nombre prédéfini d’e-mails à partir d’un compte de messagerie.

- A partir de la Palette, cliquez-déposez un composant tPOP dans l’éditeur graphique,
- Double-cliquez sur le tPOP pour afficher la vue Basic settings et paramétrer le composant.
• Renseignez les informations de connexion au serveur de messagerie : le nom de l’hôte ou son adresse IP entre guillemets dans le champ **Host** et le numéro de port dans le champ **Port**.

• Saisissez ensuite les informations d’authentification de votre compte de messagerie. Dans le champ **Username**, saisissez entre guillemets votre identifiant et dans le champ **Password**, saisissez votre mot de passe. Dans cet exemple, le serveur de messagerie s’appelle **Free**.

![tPOP_1](image)

• Dans le champ **Output directory**, sélectionnez manuellement le chemin d’accès ou cliquez sur [...] pour sélectionner le dossier de destination des e-mails récupérés sur le serveur.

• Dans le champ **Filename pattern**, définissez la syntaxe de nommage des fichiers de destination ou appuyez sur **Ctrl+Espace** pour afficher la liste des modèles prédéfinis. Dans cet exemple, la syntaxe est la suivante :

```
TalendDate.getDate("yyyyMMdd-hhmmss") + "_" + (counter_tPOP_1 + 1) + ".txt"
```

Les fichiers de destination, enregistrés au format texte, sont définis par leur date, heure et ordre de collecte.

• Décochez la case **Retrieve all emails?** pour ne récupérer qu’un nombre défini d’e-mails. Dans cet exemple, 10 messages sont récupérés.

• Cochez la case **Delete emails from server** pour supprimer les e-mails du serveur une fois qu’ils ont été récupérés en local.

• Sélectionnez un type de protocole à partir de la liste déroulante **Choose the protocol**. Le choix du protocole dépend de celui utilisé par votre fournisseur de messagerie. Certains fournisseurs comme **Gmail** utilisent les deux protocoles. Dans cet exemple, le protocole sélectionné est **pop3**.

• Enregistrez le Job puis appuyez sur **F6** pour l’exécuter.
Le composant **tPOP** a ainsi récupéré les 10 derniers messages du compte de messagerie spécifié.

Pour chaque e-mail récupéré, un fichier .txt a été créé dans le dossier **tPOP**. Chaque fichier contient les métadonnées d’envoi du message (expéditeur, destinataire, objet du message) ainsi que les données propres au message.
**tPostgresPlusBulkExec**

Ce composant améliore les performances pendant les opérations d’Insert dans une base de données DB2.

Le **PostgresPlusBulkExec** exécute des opérations d’Insert sur les données fournies.

Les composants tPostgresPlusOutputBulk et tPostgresPlusBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tPostgresPlusOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans la base de données.

**Propriétés du tPostgresPlusBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tPostgresPlusBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresPlusBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou Repository</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Remarque :</strong></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par</td>
</tr>
</tbody>
</table>
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Liste des versions de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification sur l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois.</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom du fichier à charger.</td>
</tr>
</tbody>
</table>

**Avertissement :**

Ce fichier est situé sur la machine spécifiée par l’URI dans le champ Host et doit être sur l
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local **(Built-in)**, soit distant dans le **Repository**.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. | a même machine que le serveur de la base de données. |

| **Built-In** | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**. |

| **Repository** | Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.  
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.  
Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com). |

| **Advanced settings** |
| **Action** | Sélectionnez l'opération que vous voulez effectuer : **Bulk insert ou Bulk update**. Les informations demandées seront différentes en fonction de l'action choisie. |

| **Field terminated by** | Caractère, chaîne ou expression régulière séparant les champs. |

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |
### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes PostgresPlus.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé au composant **tPostgresPlusBulkExec**, consultez :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du **tMysqlOutputBulkExec**.
- Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du **tOracleBulkExec**.
**tPostgresPlusClose**

Ce composant permet de fermer une connexion à la base de données PostgresPlus.

**Propriétés du tPostgresPlusClose Standard**

Ces propriétés sont utilisées pour configurer le tPostgresPlusClose s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresPlusClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant <strong>tPostgresPlusConnection</strong> dans la liste.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
Ce composant utilise une connexion unique pour commiter en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

Le tPostgresPlusCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tPostgresPlusCommit Standard

Ces propriétés sont utilisées pour configurer le tPostgresPlusCommit s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresPlusCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

| Database | Sélectionnez un type de base de données dans la liste et cliquez sur Apply. |
| Component list | S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tPostgresPlusConnection dans la liste. |
| Close connection | Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. Si vous utilisez un lien de type Row > Main pour relier le tPostgresPlusCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants PostgresPlus et notamment avec les composants tPostgresPlusConnection et tPostgresPlusRollback.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour plus d’informations relatives au fonctionnement du composant tPostgresPlusCommit, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tPostgresPlusConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Le tPostgresPlusConnection ouvre une connexion vers une base de données afin d'effectuer une transaction.

Propriétés du tPostgresPlusConnection Standard

Ces propriétés sont utilisées pour configurer le tPostgresPlusConnection s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresPlusConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Liste des versions de la base de données.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets.</td>
</tr>
</tbody>
</table>
If you are using a PostgresPlus database connection, you need to check the box and click **OK** to save the settings.

### Use or register a shared DB Connection

Check this box to share your database connection or retrieve a shared connection from a parent or child job. In the **Shared DB Connection Name** field, enter a name for the shared database connection. This allows you to share a database connection (except for the database schema parameter) with multiple connection components, at different levels of jobs, children or parents.

This option is incompatible with the options **Use dynamic job** and **Use an independent process to run subjob** of component **tRunJob**. Using a shared connection with a component **tRunJob** with one of these options activated will cause your job to fail.

### Advanced settings

#### Auto Commit

Check this box to automatically commit any modifications to the database when the transaction is completed.

When this box is checked, you cannot use the commit component to explicitly commit the modifications to the database. Similarly, when using a commit component, this box must be unchecked. By default, the auto-commit feature is disabled, and modifications must be committed explicitly.

Note that the auto-commit feature allows each SQL instruction to be committed as a unique transaction immediately after its execution, and the commit component does not commit until all instructions are executed. For this reason, if you need more space to manage your transactions in a job, it is recommended to use a commit component.

### Utilisation

#### Rule of use

This component is generally used with PostgresPlus components and especially with components **tPostgresPlusCommit** and **tPostgresPlusRollback**.
Scénario associé

Pour plus d’informations relatives au fonctionnement du composant tPostgresPlusConnection, consultez tMysqlConnection à la page 2618.
tPostgresPlusInput

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

Le tPostgresPlusInput lit une base de données et en extrait des champs à l’aide de requêtes.

**Propriétés du tPostgresPlusInput Standard**

Ces propriétés sont utilisées pour configurer le tPostgresPlusInput s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresPlusInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue <strong>Basic settings</strong> du composant.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong> Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une</td>
</tr>
</tbody>
</table>
connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Liste des versions de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
  • **View schema** : sélectionnez cette option afin de voir le schéma.
  • **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
  • **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. |
### Built-in
Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository

<table>
<thead>
<tr>
<th>Table name</th>
<th>Nom de la table de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query type et Query</strong></td>
<td>Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>Use cursor</th>
<th>Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d’optimiser les performances</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trim all the String/Char columns</strong></td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td><strong>Trim column</strong></td>
<td>Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>NB_LINE</strong> : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <em>After</em> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>QUERY</strong> : requête traitée. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

### Utilisation

| Règle d’utilisation | Ce composant couvre toutes les possibilités de requête SQL dans les bases de données PostgresPlus. |
Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénarios associés

Pour un scénario associé, consultez :
tPostgresPlusOutput

Ce composant exécute l’action définie sur la table et/ou sur les données contenues dans la table, à partir du flux entrant du composant précédent dans le Job.

Le tPostgresPlusOutput écrit, met à jour, effectue les changements ou supprime les entrées dans une base de données.

Propriétés du tPostgresPlusOutput Standard

Ces propriétés sont utilisées pour configurer le tPostgresPlusOutput s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresPlusOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>![Icone de connexion]</td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>![Icone de connexion]</td>
<td>Remarque : Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par</td>
</tr>
</tbody>
</table>
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Liste des versions de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit déjà exister pour que l’opération d’Insert soit autorisée.</td>
</tr>
</tbody>
</table>
| Action on table     | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :

   - **None** : n’effectuer aucune opération de table.
   - **Drop and create table** : supprimer la table puis en créer une nouvelle.
   - **Create table** : créer une table qui n’existe pas encore.
   - **Create table if doesn’t exist** : créer la table si nécessaire.
   - **Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.
   - **Clear table** : supprimer le contenu de la table.
   - **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| Action on data      | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

   - **Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons. |
**Update** : Mettre à jour les entrées existantes.

**Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.

**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

**Delete** : Supprimer les entrées correspondantes au flux d’entrée.

**Avertissement** :
Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations Update et Delete. Pour cela, cliquez sur le bouton [...] à côté du champ Edit Schema et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet Advanced settings pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case Use field options et sélectionnez la case Key in update correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases Key in delete pour les opérations de suppression (Delete).

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

**Die on error**
Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects.

### Advanced settings

<table>
<thead>
<tr>
<th>Commit every</th>
<th>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional Columns</td>
<td>Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</td>
</tr>
<tr>
<td>Name</td>
<td>Saisissez le nom de la colonne à modifier ou à insérer.</td>
</tr>
<tr>
<td>SQL expression</td>
<td>Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.</td>
</tr>
<tr>
<td>Position</td>
<td>Sélectionnez Before, Replace ou After, en fonction de l’action à effectuer sur la colonne de référence.</td>
</tr>
<tr>
<td>Reference column</td>
<td>Saisissez une colonne de référence que le composant tPostgresqlOutput peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.</td>
</tr>
<tr>
<td>Use field options</td>
<td>Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.</td>
</tr>
<tr>
<td>Enable debug mode</td>
<td>Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.</td>
</tr>
</tbody>
</table>
**Support null in “SQL WHERE” statement**
Cochez cette case pour prendre en compte les valeurs Null d’une table de base de données.

*i* Remarque :
Assurez-vous que la case **Nullable** est bien cochée pour les colonnes du schéma correspondantes.

**Use Batch**
Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.

*i* Remarque :
Cette case est disponible lorsque vous sélectionnez **Insert**, **Update** ou **Delete** dans la liste **Action on data**.

**Batch Size**
Spécifiez le nombre d’enregistrements à traiter dans chaque lot.

Ce champ est disponible uniquement lorsque la case **Use batch mode** est cochée.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

| **NB_LINE** | nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier. |
| **NB_LINE_UPDATED** | nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier. |
| **NB_LINE_INSERTED** | nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier. |
| **NB_LINE_DELETED** | nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier. |
| **NB_LINE_REJECTED** | nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier. |
| **QUERY** | requête traitée. Cette variable est une variable **After** et retourne une chaîne de caractères. |

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données PostgresPlus. Il permet aussi de créer un flux de rejet avec un lien Row &gt; Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  
La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. 

Scénarios associés

Pour des scénarios associés, consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667 du composant tMysqlOutput.
tPostgresPlusOutputBulk

Prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Postgresql.

Le tPostgresPlusOutputBulk écrit un fichier composé de colonnes et basé sur le séparateur défini et sur les standards Postgresql.

Les composants tPostgresPlusOutputBulk et tPostgresPlusBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tPostgresPlusOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Propriétés du tPostgresPlusOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tPostgresPlusOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresPlusOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom du fichier à générer.</td>
</tr>
<tr>
<td></td>
<td><strong>Avertissement</strong> :</td>
</tr>
<tr>
<td></td>
<td>Ce fichier est généré sur la machine locale ou dans un dossier partagé sur le réseau local.</td>
</tr>
<tr>
<td>Append</td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
</tbody>
</table>
## Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Built-In

Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository

Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

## Advanced settings

| **Row separator** | Chaîne (ex : \n sous Unix) séparant les lignes. |
| **Field separator** | Caractère, chaîne ou expression régulière séparant les champs. |
| **Include header** | Cochez cette case pour inclure l’en-tête des colonnes dans le fichier. |
| **Encoding** | Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |
Global Variables

| Global Variables | NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. Une variable *Flow* fonctionne durant l’exécution d'un composant. Une variable *After* fonctionne après l’exécution d’un composant. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec le composant *tPostgresPlusBulkExec*. Ensemble ils offrent un gain de performance important pour l’alimentation d’une base de données PostgresPlus. |

Scénarios associés

Pour un scénario associé au composant *tPostgresPlusOutputBulk*, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du *tMysqlOutputBulk*.
- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du *tMysqlOutputBulkExec*.
- **Scénario : Supprimer et insérer des données dans une base Oracle** à la page 2914 du *tOracleBulkExec*. 

tPostgresPlusOutputBulkExec

Ce composant améliore les performances pendant les opérations d’Insert dans une base de données PostgresPlus.

Le tPostgresPlusOutputBulkExec effectue une action d’Insert sur les données fournies.

Les composants tPostgresPlusOutputBulk et tPostgresPlusBulkExec sont généralement utilisés ensemble comme deux parties d’un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l’opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tPostgresPlusOutputBulkExec.

Propriétés du tPostgresPlusOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tPostgresPlusOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresPlusOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Liste des versions de la base de données.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Seuls localhost, 127.0.0.1 ou l’adresse IP exacte de la machine locale permettent un fonctionnement optimal.</td>
</tr>
<tr>
<td></td>
<td>Le serveur de la base de données doit être installé sur la même machine que le Studio Talend ou le Job contenant un tPostgresPlusOutputBulkExec.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Nom exact du schéma.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit déjà exister pour que l’opération d’Insert soit autorisée.</td>
</tr>
</tbody>
</table>
| **Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
    - **None** : n’effectuer aucune opération de table.  
    - **Drop and create the table** : supprimer la table puis en créer une nouvelle.  
    - **Create a table** : créer une table qui n’existe pas encore.  
    - **Create table if doesn’t exist** : créer la table si nécessaire.  
    - **Clear a table** : supprimer le contenu de la table. |
| **File Name** | Nom du fichier à générer et à charger.  
    - **Avertissement** :  
        Ce fichier est généré sur la machine spécifiée par l’URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.  
    Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
    - **View schema** : sélectionnez cette option afin de voir le schéma.  
    - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
    - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. |
| **Built-In** | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*. |
| **Repository** | Le schéma existe déjà et il est stocké dans le *Repository*. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*. Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets. Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center ([https://help.talend.com](https://help.talend.com)). |

### Advanced settings

| **Action** | Sélectionnez l’opération que vous voulez effectuer : **Bulk insert** ou **Bulk update**. Les informations demandées seront différentes en fonction de l’action choisie. |
| **File type** | Sélectionnez le type de fichiers à traiter. |
| **Null string** | Chaîne de caractères affichée pour signifier que la valeur est nulle. |
| **Row separator** | Chaîne (ex : "\n" sous Unix) séparant les lignes. |
| **Fields terminated by** | Caractère, chaîne ou expression régulière séparant les champs :  
- **None** : aucun caractère de séparation.  
- **Whitespace** : le caractère de séparation est un espace.  
- **EOF (used for loading LOBs from lobfile)** : la séparation est constituée d’un caractère de fin de fichier (End Of File).  
- **Other terminator** : la séparation est constituée d’un caractère autre que ceux précédemment énoncés. |
| **Escape character** | Caractère d’échappement de la ligne. |
| **Text enclosure** | Caractère utilisé pour entourer le texte. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

### Utilisation

| **Règle d’utilisation** | Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans la base de données. |
Scénarios associés

Pour un scénario associé au composant **tPostgresPlusOutputBulkExec**, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du **tMysqlOutputBulk**.
- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du **tMysqlOutputBulkExec**.
- **Scénario : Supprimer et insérer des données dans une base Oracle** à la page 2914 du **tOracleBulkExec**.
tPostgresPlusRollback

Ce composant permet d’éviter le commit de transaction involontaire.
Le tPostgresPlusRollback annule la transaction dans une base de données connectée.

Propriétés du tPostgresPlusRollback Standard

Ces propriétés sont utilisées pour configurer le tPostgresPlusRollback s’exécutant dans le framework de Jobs Standard.
Le composant tPostgresPlusRollback Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tPostgresPlusConnection dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics   | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec d’autres composants PostgresPlus, notamment les composants tPostgresPlusConnection et tPostgresPlusCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes...</td>
</tr>
</tbody>
</table>
bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de Paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

tPostgresPlusRow

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler), selon la nature de la requête et de la base de données. Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Le tPostgresPlusRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Propriétés du tPostgresPlusRow Standard

Ces propriétés sont utilisées pour configurer le tPostgresPlusRow s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresPlusRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Liste des versions de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma</td>
</tr>
</tbody>
</table>

**Username et Password**

Informations d’authentification de l’utilisateur de base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Table name</th>
<th>Nom de la table de base à lire.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.</td>
</tr>
<tr>
<td>Query</td>
<td>Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Rejects.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Propagate QUERY’s recordset</th>
<th>Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Remarque</strong> :</td>
</tr>
<tr>
<td></td>
<td>Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type Object. Ce composant est généralement suivi du tParseRecordSet.</td>
</tr>
<tr>
<td></td>
<td><strong>Parameter Index</strong> : Saisissez la position du paramètre dans l'instruction SQL.</td>
</tr>
<tr>
<td></td>
<td><strong>Parameter Type</strong> : Saisissez le type du paramètre.</td>
</tr>
<tr>
<td></td>
<td><strong>Parameter Value</strong> : Saisissez la valeur du paramètre.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque</strong> :</td>
</tr>
<tr>
<td></td>
<td>Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.</td>
</tr>
<tr>
<td>Commit every</td>
<td>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

|---|---|---|

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsqu’un paramètre de votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les</td>
</tr>
</tbody>
</table>
Scénarios associés

Pour un scénario associé, consultez :

- **Scénario : Combiner deux flux pour une sortie sélective** à la page 2706.
- **Procédure** du composant **tDBSQLRow**.
- **Scénario : Supprimer et re-générer un index de table MySQL** à la page 2700 du composant **tMysqlRow**.
tPostgresPlusSCD

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée. Le tPostgresPlusSCD reflète et traque les modifications d’une table SCD PostgresPlus dédiée.

**Propriétés du tPostgresPlusSCD Standard**

Ces propriétés sont utilisées pour configurer le tPostgresPlusSCD s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresPlusSCD Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.
<table>
<thead>
<tr>
<th><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td><strong>DB Version</strong></td>
</tr>
<tr>
<td><strong>Port</strong></td>
</tr>
<tr>
<td><strong>Database</strong></td>
</tr>
<tr>
<td><strong>Schema</strong></td>
</tr>
</tbody>
</table>
| **Username** et **Password** | Informations d'authentification de l'utilisateur de la base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Table** | Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois. |
| **Schema** et **Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
  - **View schema** : sélectionnez cette option afin de voir le schéma.  
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [**Repository Content**].  
| **Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**. |
| **Repository** : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**. |
SCD Editor

L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension.

Pour plus d’informations, consultez Méthodologie de gestion du SCD à la page 2716.

Use memory saving Mode

Cochez cette case pour améliorer les performances du système.

Source keys include Null

Cochez cette case pour permettre aux colonnes source clés de contenir des valeurs nulles.

Avertissement :

Lorsque cette option est cochée, vous devez faire attention à l’unicité des clé(s) source.

Die on error

Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

Advanced settings

End date time details

Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format HH:mm:ss. La valeur par défaut pour ce champ est 12:00:00.

Ce champ apparaît uniquement lorsqu’un SCD de Type 2 est utilisé et lorsque Fixed year value est sélectionné pour créer la date de fin du SCD.

Debug mode

Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

Variables globales

Global Variables

NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.

NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.

NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Limitation</td>
<td>Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé, consultez tMysqlSCD à la page 2712.
tPostgresPlusSCDELT

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table PostgresPlus SCD dédiée.

Le tPostgresPlusSCDELT reflète et traque les modifications d’une table PostgresPlus SCD dédiée.

Propriétés du tPostgresPlusSCDELT Standard

Ces propriétés sont utilisées pour configurer le tPostgresPlusSCDELT s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresPlusSCDELT Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Liste des versions de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d'authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Source table</td>
<td>Nom de la table contenant les données à filtrer.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu'une seule table peut être écrite à la fois pour que l'opération d'insert soit autorisée.</td>
</tr>
</tbody>
</table>
| Action on table    | Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :
|                    | **None** : n'effectuer aucune opération de table. |
|                    | **Drop and create the table** : supprimer la table puis en créer une nouvelle. |
|                    | **Create a table** : créer une table qui n'existe pas encore. |
|                    | **Create table if doesn’t exist** : créer la table si nécessaire. |
|                    | **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. |
|                    | **Clear a table** : supprimer le contenu de la table. |
|                    | **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs. |
|                    | Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : |
| **Surrogate Key** | Sélectionnez dans la liste une colonne à utiliser comme clé de substitution. |
| **Creation** | Sélectionnez la méthode à utiliser pour générer la clé de substitution.  
Pour plus d’informations concernant les méthodes de création, consultez [Méthodologie de gestion du SCD](#) à la page 2716. |
| **Source Keys** | Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l’unicité des données entrantes. |
| **Use SCD Type 1 fields** | Utilisez le type 1 si vous n’avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications. |
| **Use SCD Type 2 fields** | Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.  
**Start date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l’une des colonnes d’entrée du schéma comme date de départ (Start Date) dans la table SCD.  
**End Date** : Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne **End Date** a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l’option **Fixed Year value** et saisir une année fictive.  
**Log Active Status** : Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut **true** et **false**. |
| **Built-in** | Le schéma est créé et conservé pour ce composant seulement. Voir également le [Guide utilisateur du Studio Talend](#). |
| **Repository** | Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le [Guide utilisateur du Studio Talend](#). |

- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
Cette colonne permet de repérer facilement le journal actif.

**Log versions** : Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Debug mode</strong></th>
<th>Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th><strong>Global Variables</strong></th>
<th><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</th>
</tr>
</thead>
</table>

### Utilisation

<table>
<thead>
<tr>
<th><strong>Règle d’utilisation</strong></th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type <strong>Row Main</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable.</td>
</tr>
</tbody>
</table>

---

3240
Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Scénario associé

Pour un scénario associé, consultez tMysqlSCD à la page 2712.
tPostgresqlBulkExec

Ce composant améliore les performances lors du traitement des données de la base de données Postgresql.

Le tPostgresqlBulkExec effectue une opération Insert sur les données fournies.

Les composants tPostgresqlOutputBulk et tPostgresqlBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tPostgresqlOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Propriétés du tPostgresqlBulkExec Standard

Ces propriétés sont utilisées pour configurer le tPostgresqlBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Remarque :</td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
</tbody>
</table>
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Liste des versions de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit exister pour que l’opération d’Insert soit autorisée.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
**None** : n’effectuer aucune opération de table.  
**Drop and create the table** : supprimer la table puis en créer une nouvelle.  
**Create table** : créer une table qui n’existe pas encore.  
**Create table if not exists** : créer la table si nécessaire.  
**Drop table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.  
**Clear a table** : supprimer le contenu de la table.  
**Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| File Name  | Nom du fichier à charger.  
**Avertissement** :  
Le fichier est situé sur la machine spécifiée par l’URI dans le champ **Host** et doit être sur la même machine que le serveur de la base de données. |
### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Advanced settings

**Action on data**

Sélectionnez l’opération que vous voulez effectuer :

- **Bulk insert** : ajoute de nombreuses entrées dans la table.
- **Bulk update** : effectue des modifications simultanées dans différentes entrées.

**Copy the OID for each row**

Récupère les identifiants d’objet pour chaque ligne.

**Contains a header line with the names of each column in the file**

Spécifiez que la table contient des lignes d’en-tête.

**File type**

Sélectionnez le type de fichiers à traiter.

**Null string**

Chaîne de caractères affichée pour signifier que la valeur est nulle.
<table>
<thead>
<tr>
<th><strong>Fields terminated by</strong></th>
<th>Caractère, chaîne ou expression régulière séparant les champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Escape char</strong></td>
<td>Caractère d’échappement de la ligne.</td>
</tr>
<tr>
<td><strong>Text enclosure</strong></td>
<td>Caractères utilisés pour entourer le texte.</td>
</tr>
<tr>
<td><strong>Use standard_conforming_string ON</strong></td>
<td>Activez la variable.</td>
</tr>
<tr>
<td><strong>Force not null for columns</strong></td>
<td>Définissez la nullabilité des colonnes.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Utilisation**

**Règle d’utilisation**

Ce composant doit être utilisé en association avec le composant *tPostgresqlOutputBulk*. Ensemble, ils permettent d’obtenir des gains de performances dans une base de données Postgresql.

**Dynamic settings**

Cliquez sur le bouton `[+]` pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez *Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte* à la page 2641 et *Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement* à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*. 
Scénarios associés

Pour d'autres exemples d'utilisation du composant `tPostgresqlBulkExec`, consultez les scénarios suivants dans :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du `tMysqlOutputBulk`,

- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du `tMysqlOutputBulkExec`,

- **Scénario : Supprimer et insérer des données dans une base Oracle** à la page 2914 du `tOracleBulkExec`. 
tPostgresqlClose

Ce composant ferme une connexion à la base de données Postgresql.

**Propriétés du tPostgresqlClose Standard**

Ces propriétés sont utilisées pour configurer le tPostgresqlClose s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tPostgresqlConnection dans la liste.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tPostgresqlCommit**

Ce composant utilise une connexion unique pour commiter en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

Le tPostgresqlCommit valide les données traitées dans un Job à partir d’une base de données connectée.

**Propriétés du tPostgresqlCommit Standard**

Ces propriétés sont utilisées pour configurer le tPostgresqlCommit s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant <strong>tPostgresqlConnection</strong> dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
</tbody>
</table>
| Close connection | Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. **Avertissement** :

Si vous utilisez un lien de type Row>Main pour relier le tPostgresqlCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants PostgreSQL, notamment les composants <code>tPostgresqlConnection</code> et <code>tPostgresqlRollback</code>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton <code>[+]</code> pour ajouter une ligne à la table. Dans le champ <code>Code</code>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

## Scénario associé

Pour plus d’informations relatives au fonctionnement du composant `tPostgresqlCommit`, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tPostgresqlConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Le tPostgresqlConnection ouvre une connexion vers une base de données afin d’effectuer une transaction.

**Propriétés du tPostgresqlConnection Standard**

Ces propriétés sont utilisées pour configurer le tPostgresqlConnection s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Liste des versions de la base de données.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets</td>
</tr>
</tbody>
</table>
Use or register a shared DB Connection

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.

Cette case est indisponible lorsque la case **Specify a data source alias** est cochée.

Specify a data source alias

Cochez cette case et spécifiez l’alias de la source de données créée dans **Talend Runtime** pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans **Talend Runtime**. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans **Talend Runtime** pour récupérer les données d’une base de données MySQL à la page 2647.

Cette option est indisponible lorsque la case **Use an existing connection** est cochée.

Data source alias

Saisissez l’alias de la source de données créée du côté de **Talend Runtime**.

Ce champ est disponible uniquement lorsque la case **Specify a data source alias** est cochée.

Advanced settings

**Auto Commit**

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commises de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu’à ce
que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec des composants Postgresql, notamment les composants **tPostgresqlCommit** et **tPostgresqlRollback**. |

**Scénario associé**

Pour un scénario associé au composant **tPostgresqlConnection**, consultez **tMysqlConnection** à la page 2618.
tPostgresqlInput

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

Le tPostgresqlInput lit une base de données et en extrait des champs à l'aide de requêtes.

Propriétés du tPostgresqlInput Standard

Ces propriétés sont utilisées pour configurer le tPostgresqlInput s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>![Icon]</td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>
| ![Icon] | Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une |
connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Liste des versions de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (**Built-in**), soit distant dans le **Repository**.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. |
**Built-in**: Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


<table>
<thead>
<tr>
<th>Table Name</th>
<th>Nom de la table à lire.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query type et Query</strong></td>
<td>Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.</td>
</tr>
</tbody>
</table>

**Specify a data source alias**

Cochez cette case et spécifiez l’alias de la source de données créée dans Talend Runtime pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647.

Cette option est indisponible lorsque la case Use an existing connection est cochée.

| Data source alias | Saisissez l’alias de la source de données créée du côté de Talend Runtime. Ce champ est disponible uniquement lorsque la case Specify a data source alias est cochée. |

**Advanced settings**

**Use cursor**

Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d’optimiser les performances.

**Trim all the String/Char columns**

Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.

**Trim column**

Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. **QUERY** : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Postgresql.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton Install dans l’onglet Component. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet Modules de la perspective Integration de votre studio. Vous pouvez</td>
</tr>
</tbody>
</table>
Scénarios associés

Pour un scénario associé, consultez les scénarios du composant **tPostgresqlInput** :

- **Scénario :** Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520.
tPostgresqlOutput

Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Le tPostgresqlOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

**Propriétés du tPostgresqlOutput Standard**

Ces propriétés sont utilisées pour configurer le tPostgresqlOutput s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Liste des versions de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée : <strong>None</strong> : n’effectuer aucune opération de table. <strong>Drop and create table</strong> : supprimer la table puis en créer une nouvelle. <strong>Create table</strong> : créer une table qui n’existe pas encore. <strong>Create table if not exists</strong> : créer la table si nécessaire. <strong>Drop table if exists and create</strong> : supprimer la table si elle existe déjà, puis en créer une nouvelle. <strong>Clear table</strong> : supprimer le contenu de la table. <strong>Truncate table</strong> : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.</td>
</tr>
<tr>
<td>Action on data</td>
<td>Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : <strong>Insert</strong> : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons. <strong>Update</strong> : Mettre à jour les entrées existantes.</td>
</tr>
</tbody>
</table>
**Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.

**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

**Delete** : Supprimer les entrées correspondantes au flux d’entrée.

⚠️ **Avertissement** :
Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...], à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Use spatial options

**Remarque :**
Cette case est disponible lorsque vous sélectionnez l’option de création de table dans la liste *Action on table*.

Cochez cette case pour utiliser la base de données PostgreSQL en tant que base de données spatiale pour un système d’informations géographique (SIG). Lorsque cette case est cochée, deux autres cases s’affichent :

- **Create Spatial index** : cochez cette case pour créer un index spatial.
- **Create geometry columns reference** : cochez cette case pour créer la référence des colonnes de type Geometry.

Pour plus d’informations concernant la base de données PostgreSQL, consultez [http://www.postgresql.org/about/](http://www.postgresql.org/about/).

### Specify a data source alias

Cochez cette case et spécifiez l’alias de la source de données créée dans *Talend Runtime* pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans *Talend Runtime*. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans *Talend Runtime* pour récupérer les données d’une base de données MySQL à la page 2647.

Cette option est indisponible lorsque la case *Use an existing connection* est cochée.

### Data source alias

Saisissez l’alias de la source de données créée du côté de *Talend Runtime*.

Ce champ est disponible uniquement lorsque la case Specify a data source alias est cochée.

### Die on error

Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.
### Advanced settings

<table>
<thead>
<tr>
<th><strong>Commit every</strong></th>
<th>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Additional Columns</strong></td>
<td>Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Saisissez le nom de la colonne à modifier ou à insérer.</td>
</tr>
<tr>
<td><strong>SQL expression</strong></td>
<td>Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.</td>
</tr>
<tr>
<td><strong>Position</strong></td>
<td>Sélectionnez Before, Replace ou After, en fonction de l’action à effectuer sur la colonne de référence.</td>
</tr>
<tr>
<td><strong>Reference column</strong></td>
<td>Saisissez une colonne de référence que le composant \texttt{tPostgresqlOutput} peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.</td>
</tr>
<tr>
<td><strong>Use save point</strong></td>
<td>Cochez cette case pour utiliser des points de sauvegarde au cours de la transaction. Cette case est indisponible si vous cochez :</td>
</tr>
<tr>
<td></td>
<td>• la case \texttt{Die on error} dans la vue \texttt{Basic settings}, ou</td>
</tr>
<tr>
<td></td>
<td>• la case \texttt{Use Batch Size} dans la vue \texttt{Advanced settings}.</td>
</tr>
<tr>
<td></td>
<td>Cette case ne fonctionne pas si vous :</td>
</tr>
<tr>
<td></td>
<td>• saisissez 0 dans le champ \texttt{Commit every}, ou</td>
</tr>
<tr>
<td></td>
<td>• cochez la case \texttt{Use an existing connection} dans la vue \texttt{Basic settings} et que le mode \texttt{Auto Commit} est activé dans le composant de connexion à la base de données.</td>
</tr>
<tr>
<td><strong>Use field options</strong></td>
<td>Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.</td>
</tr>
<tr>
<td><strong>Enable debug mode</strong></td>
<td>Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.</td>
</tr>
<tr>
<td><strong>Support null in “SQL WHERE” statement</strong></td>
<td>Cochez cette case pour prendre en compte les valeurs Null d’une table de base de données.</td>
</tr>
</tbody>
</table>

**Remarque :**
Assurez-vous que la case Nullable est bien cochée pour les colonnes du schéma correspondantes.

<table>
<thead>
<tr>
<th>Use Batch</th>
<th>Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remarque</strong></td>
<td>Cette case est disponible uniquement lorsque vous sélectionnez Insert, Update ou Delete dans la liste Action on data.</td>
</tr>
<tr>
<td><strong>Batch Size</strong></td>
<td>Spécifiez le nombre d’enregistrements à traiter dans chaque lot.</td>
</tr>
<tr>
<td></td>
<td>Ce champ est disponible uniquement lorsque la case Use batch mode est cochée.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong> : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong> : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong> : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong> : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_REJECTED</strong> : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>QUERY</strong> : requête traitée. Cette variable est une variable After et retourne une chaîne de caractères.</td>
<td></td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
<td>Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
Utilisation

| Règle d'utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données Postgresql. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénarios associés

Pour un scénario associé au composant tPostgresqlOutput consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667.
tPostgresqlOutputBulk

Prépare le fichier à utiliser comme paramètres dans la requête INSERT servant à alimenter une base de données Postgresql.

Le tPostgresqlOutputBulk écrit un fichier composé de colonnes et basé sur le séparateur défini et sur les standards Postgresql.

Les composants tPostgresqlOutputBulk et tPostgresqlBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tPostgresqlOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Propriétés du tPostgresqlOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tPostgresqlOutputBulk s'exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom du fichier à générer.</td>
</tr>
<tr>
<td>Avertissement</td>
<td>Le fichier est généré sur la machine locale ou dans un fichier partagé sur le réseau local.</td>
</tr>
<tr>
<td>Append</td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
</tbody>
</table>
### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Advanced settings

- **Row separator** : Chaîne (ex : "\n" sous Unix) séparant les lignes.
- **Field separator** : Caractère, chaîne ou expression régulière séparant les champs.
- **Include header** : Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.
- **Encoding** : Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.
- **tStatCatcher Statistics** : Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>nb_line : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>error_message : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable flow fonctionne durant l’exécution d’un composant. Une variable after fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec le composant tPostgresqlBulkExec. Ensemble ils offrent un gain de performance important pour l’alimentation d’une base de données Postgresql. |

Scénarios associés

Pour un scénario associé au composant tPostgresqlOutputBulk, consultez :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du tMysqlOutputBulk.
- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec.
- Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du tOracleBulkExec.
tPostgresqlOutputBulkExec

Ce composant améliore les performances pendant les opérations d’Insert dans une base de données Postgresql.

Le tPostgresqlOutputBulkExec effectue une action d’Insert sur les données fournies.

Les composants tPostgresqlOutputBulk et tPostgresqlBulkExec sont généralement utilisés ensemble comme deux parties d’un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l’opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tPostgresqlOutputBulkExec.

Propriétés du tPostgresqlOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tPostgresqlOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Liste des versions de la base de données.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>localhost, 127.0.0.1</td>
<td>Le serveur de base de données doit être installé sur la même machine que le Studio Talend ou le Job comprenant le tPostgresqlOutputBulkExec.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit déjà exister pour que l’opération d’Insert soit autorisée.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
**None**: n’effectuer aucune opération de table.  
**Drop and create table**: supprimer la table puis en créer une nouvelle.  
**Create table**: créer une table qui n’existe pas encore.  
**Create table if not exists**: créer la table si nécessaire.  
**Drop table if exists and create**: supprimer la table si elle existe déjà, puis en créer une nouvelle.  
**Clear a table**: supprimer le contenu de la table. |
| File Name | Nom du fichier à générer et à charger.  
**Avertissement** : Le fichier est généré sur la machine spécifiée par l’URI dans le champ Host et doit être sur la même machine que le serveur de la base de données. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la
métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

**Repository** : Le schéma existe déjà et il est stocké dans le *Repository*. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

### Advanced settings

| Action on data | Sélectionnez l’opération que vous voulez effectuer :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bulk insert</strong></td>
<td>ajoute différentes entrées à la table. Si des doublons sont trouvés, le Job s’arrête.</td>
</tr>
<tr>
<td><strong>Bulk update</strong></td>
<td>effectue des modifications simultanées dans différentes entrées.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Copy the OID for each row</th>
<th>Récupère les identifiants d’objet pour chaque ligne.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains a header line with the names of each column in the file</td>
<td>Spécifiez que la table contient des lignes d’en-tête.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Sélectionnez l’encodage à partir de la liste ou sélectionnez <em>Custom</em> et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>File type</td>
<td>Sélectionnez le type de fichiers à traiter.</td>
</tr>
<tr>
<td>Null string</td>
<td>Chaîne de caractères affichée pour signifier que la valeur est nulle.</td>
</tr>
<tr>
<td>Row separator</td>
<td>Chaîne (ex : &quot;\n&quot; sous Unix) séparant les lignes.</td>
</tr>
<tr>
<td>Fields terminated by</td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td>Escape char</td>
<td>Caractère d’échappement de la ligne.</td>
</tr>
<tr>
<td>Text enclosure</td>
<td>Caractères utilisés pour entourer le texte.</td>
</tr>
<tr>
<td><strong>Activate standard_conforming_string</strong></td>
<td>Activez le paramètre standard_conforming_string pour traiter les backslashes comme une chaîne ordinaire et non pas comme un caractère d’échappement.</td>
</tr>
</tbody>
</table>
### Force not null for columns
Définissez la nullabilité des colonnes.

**Force not null**: Cochez la case correspondant à la colonne *(Column)* que vous souhaitez définir comme n’étant pas nulle.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Le serveur de base de données doit être installé sur la même machine que le <em>Studio Talend</em> ou le Job com prenant un <em>tPostgresqlOutputBulkExec</em> afin que le composant fonctionne correctement.</td>
</tr>
</tbody>
</table>

## Scénarios associés

Pour un scénario associé au composant *tPostgresqlOutputBulkExec*, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du *tMysqlOutputBulk*.

- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du *tMysqlOutputBulkExec*.

- **Scénario : Supprimer et insérer des données dans une base Oracle** à la page 2914 du *tOracleBulkExec*. 
tPostgresqlRollback

Ce composant évite le commit de transaction involontaire. Le tPostgresqlRollback annule la transaction dans une base de données connectée.

Propriétés du tPostgresqlRollback Standard

Ces propriétés sont utilisées pour configurer le tPostgresqlRollback s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tPostgresqlConnection dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec d’autres composants Postgresql, notamment les composants tPostgresqlConnection et tPostgresqlCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes</td>
</tr>
</tbody>
</table>
bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Pour un scénario associé au composant tPostgresqlRollback, consultez tMysqlRollback à la page 2694.
Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler), selon la nature de la requête et de la base de données. Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Le tPostgresqlRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

**Propriétés du tPostgresqlRow Standard**

Ces propriétés sont utilisées pour configurer le tPostgresqlRow s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque** :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

**Remarque** :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans...
la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend.*

| DB Version | Liste des versions de la base de données. |
| Host | Adresse IP du serveur de base de données. |
| Port | Numéro du port d’écoute du serveur de base de données. |
| Database | Nom de la base de données. |
| Schema | Nom exact du schéma. |
| **Username** et **Password** | Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (*Built-in*), soit distant dans le *Repository*. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend.*
**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Query type</th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong> : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Sélectionnez la requête appropriée dans le Repository. Le champ <em>Query</em> est renseigné automatiquement.</td>
<td></td>
</tr>
</tbody>
</table>

| Query | Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma. |

| Specify a data source alias | Cochez cette case et spécifiez l’alias de la source de données créée dans **Talend Runtime** pour utiliser le pool de connexions partagées défini dans la configuration des données source. Cette option fonctionne lorsque vous déployez et exécutez votre Job dans **Talend Runtime**. Pour voir un exemple de cas d’utilisation, consultez Scénario : Déploiement de votre Job dans **Talend Runtime** pour récupérer les données d’une base de données **MySQL** à la page 2647. Cette option est indisponible lorsque la case **Use an existing connection** est cochée. |

| Data source alias | Saisissez l’alias de la source de données créée du côté de **Talend Runtime**. Ce champ est disponible uniquement lorsque la case **Specify a data source alias** est cochée. |

| Die on error | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**. |

**Advanced settings**

| Propagate QUERY's recordset | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**. |

**Remarque** :

Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type **Object**. Ce composant est généralement suivi du **tParseRecordSet**.

Parameter Index : Saisissez la position du paramètre dans l’instruction SQL.

Parameter Type : Saisissez le type du paramètre.

Parameter Value : Saisissez la valeur du paramètre.

**Remarque :**
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

| Commit every | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

| Global Variables | QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre co
connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu'un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

### Scénarios associés

Pour un scénario associé, consultez :

- Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
- Procédure du composant tDBSQLRow.
- Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du composant tMysqlRow.
tPostgresqlSCD

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée. Le tPostgresqlSCD reflète et traque les modifications d’une table SCD Postgresql dédiée.

**Propriétés du tPostgresqlSCD Standard**

Ces propriétés sont utilisées pour configurer le tPostgresqlSCD s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlSCD Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.
<table>
<thead>
<tr>
<th><strong>Built-in</strong></th>
<th>Propriétés utilisées ponctuellement.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
</tbody>
</table>

DB Version

| Sélectionnez dans la liste la version de la base de données que vous utilisez. |

Host

| Adresse IP du serveur de base de données. |

Port

| Numéro du port d'écoute du serveur de base de données. |

Database

| Nom de la base de données. |

Schema

| Nom du schéma de la base de données. |

Username et Password

| Informations d'authentification de l'utilisateur de la base de données. |

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |

Table

| Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois. |

Schema et Edit Schema

| Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. |

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content. |

| Built-in | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |

SCD Editor

L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d’informations, consultez Méthodologie de gestion du SCD à la page 2716.

Use memory saving Mode

Cochez cette case pour améliorer les performances du système.

Source keys include Null

Cochez cette case pour permettre aux colonnes source clés de contenir des valeurs nulles.

Avertissement :

Lorsque cette option est cochée, vous devez faire attention à l’unicité des clé(s) source.

Die on error

Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

Advanced settings

End date time details

Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format HH:mm:ss. La valeur par défaut pour ce champ est 12:00:00.

Ce champ apparaît uniquement lorsqu’un SCD de Type 2 est utilisé et lorsque Fixed year value est sélectionné pour créer la date de fin du SCD.

Debug mode

Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

Variables globales

Global Variables

NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.

NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.

NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.</th>
</tr>
</thead>
</table>
| **Dynamic settings** | Cliquez sur le bouton [+1] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.  
Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**. |
| **Limitation** | Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD. |

### Scénario associé

Pour un scénario associé, consultez **tMysqlSCD** à la page 2712.
tPostgresqlSCDELT

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table Postgresql SCD dédiée.

Le tPostgresqlSCDELT reflète et traque les modifications d’une table Postgresql SCD dédiée.

**Propriétés du tPostgresqlSCDELT Standard**

Ces propriétés sont utilisées pour configurer le tPostgresqlSCDELT s’exécutant dans le framework de Jobs Standard.

Le composant tPostgresqlSCDELT Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
| **DB Version** | Sélectionnez dans la liste la version de la base de données que vous utilisez. |
| **Host** | Adresse IP du serveur de base de données. |
| **Port** | Numéro du port d’écoute du serveur de base de données. |
| **Database** | Nom de la base de données. |
| **Username et Password** | Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Source table** | Nom de la table contenant les données à filtrer. |
| **Table** | Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois pour que l’opération d’insert soit autorisée. |
| **Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée : **None** : n’effectuer aucune opération de table. **Drop and create the table** : supprimer la table puis en créer une nouvelle. **Create a table** : créer une table qui n’existe pas encore. **Create table if doesn’t exist** : créer la table si nécessaire. **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle. **Clear a table** : supprimer le contenu de la table. **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles : • **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.

- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

<table>
<thead>
<tr>
<th><strong>Built-in</strong></th>
<th>Le schéma est créé et conservé pour ce composant seulement. Voir également le <em>Guide utilisateur du Studio Talend</em>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

**Surrogate Key**

Sélectionnez dans la liste une colonne à utiliser comme clé de substitution.

**Creation**

Sélectionnez la méthode à utiliser pour générer la clé de substitution.

Pour plus d’informations concernant les méthodes de création, consultez *Méthodologie de gestion du SCD* à la page 2716.

**Source Keys**

Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l’unicité des données entrantes.

**Use SCD Type 1 fields**

Utilisez le type 1 si vous n’avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.

**Use SCD Type 2 fields**

Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.

- **Start date**: Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l’une des colonnes d’entrée du schéma comme date de départ (Start Date) dans la table SCD.

- **End Date**: Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne *End Date* a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l’option *Fixed Year value* et saisir une année fictive.

- **Log Active Status**: Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut *true* et *false*. Cette colonne permet de repérer facilement le journal actif.
**Log versions** : Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.

### Advanced settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug mode</td>
<td>Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type **Row Main**. |
| Dynamic settings    | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table **Dynamic settings** est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données**. |
Scénario associé

Pour un scénario associé, consultez tMysqlSCD à la page 2712.
tPostjob

Ce composant déclenche une action requise après l’exécution d’un Job.
Le composant tPostjob démarre l’exécution d’un post-job.

Propriétés du tPostjob Standard

Ces propriétés sont utilisées pour configurer le tPostjob s’exécutant dans le framework de Jobs Standard.
Le composant tPostjob Standard appartient à la famille Orchestration.
Le composant de ce framework est toujours disponible.

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Global Variables


Utilisation

| Règle d’utilisation | Le tPostjob est un composant de début. Il ne peut être relié au composant suivant qu’avec une connexion de type Iterate. |

| Connections | Liens de sortie (de ce composant à un autre) :
Trigger : OnComponentOk. 
Liens d’entrée (d’un autre composant à celui-ci) :
Trigger : Synchronize, Parallelize. |
Pour plus d'informations concernant les liens, consultez le Guide utilisateur du Studio Talend.

Pour plus d'informations sur le composant tPostjob, consultez le Guide utilisateur du Studio Talend.

Scénario associé

- Scénario : Gérer des fichiers avant et après l'exécution d'un Job à la page 3292
- Scénario : Créer un diagramme en barres à partir de données d'entrée à la page 356
tPrejob

Ce composant déclenche une action requise pour l'exécution d'un Job.
Le composant tPrejob démarre l'exécution d'un pré-job.

Propriétés du tPrejob Standard

Ces propriétés sont utilisées pour configurer le tPrejob s'exécutant dans le framework de Jobs Standard.
Le composant tPrejob Standard appartient à la famille Orchestration.
Le composant de ce framework est toujours disponible.

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. |

Global Variables

| Global Variables | ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

Utilisation

| Règle d'utilisation | Le tPrejob est un composant de début. Il ne peut être relié au composant suivant qu'avec une connexion de type Iterate. |
| Connections | Liens de sortie (de ce composant à un autre) : Trigger : OnComponentOk. Liens d'entrée (d'un autre composant à celui-ci) : Trigger : Synchronize, Parallelize. |
Scénario : Gérer des fichiers avant et après l'exécution d'un Job

Dans ce scénario, un pré-Job et un post-Job sont ajoutés dans le Job décrit dans Scénario 2 : Trouver des fichiers dupliqués entre deux dossiers à la page 1160 afin de gérer les fichiers avant et après l'exécution du Job principal.

Comme décrit dans le scénario mentionné précédemment, lorsque le Job principal démarre, il crée un fichier texte temporaire afin de stocker les noms des fichiers récupérés. Si le fichier temporaire spécifié existe déjà, il est écrasé.

Afin de prévenir toute perte de données, le pré-Job enregistre une copie de sauvegarde du fichier spécifié en le renommant avant que le Job principal s'exécute.

Après l'exécution du Job principal, le post-Job supprime le fichier temporaire créé par le Job et restaure le fichier de sauvegarde, s'il existe.

Ajouter et relier les composants

Procédure

1. Ouvrez le Job principal dans l’espace de modélisation graphique puis ajoutez-y un tPrejob, un tPostjob, un tFileDelete et deux tFileCopy.
2. Reliez le tPrejob au premier tFileCopy à l’aide d’un lien Trigger > On Component Ok afin de créer le pré-Job.
3. Reliez le tPostjob au tFileDelete à l’aide d’un lien Trigger > On Component Ok et reliez l’autre tFileDelete à l’autre tFileCopy afin de créer le post-Job.
4. Nommez les composants afin d’identifier plus facilement leur rôle.
Configurer les composants

Configurer le pré-Job

Pourquoi et quand exécuter cette tâche
Dans le pré-Job, le tPrejob n’a aucun paramètre ou propriété à définir. Il déclenche seulement le tFileCopy avant l’exécution du Job principal afin de renommer le fichier spécifié. Tous les paramètres nécessaires sont donc définis dans le tFileCopy.

Procédure
1. Double-cliquez sur le tFileCopy afin d’ouvrir sa vue Basic settings.

2. Dans le champ File Name, saisissez le chemin ainsi que le nom du fichier temporaire à renommer, D:/temp/tempdata.csv dans ce scénario.

3. Dans le champ Destination directory, spécifiez le chemin ou parcourez votre système vers le dossier cible. Dans cet exemple, la copie de sauvegarde est enregistrée dans le même dossier, D:/temp/.


Configurez le post-Job

Pourquoi et quand exécuter cette tâche
Dans le post-Job, le tPostjob n’a aucun paramètre ou propriété à définir. Il déclenche seulement le tFileDelete après l’exécution du Job principal afin de supprimer le fichier temporaire utilisé pour stocker les noms des fichier récupérés. Il déclenche ensuite le tFileCopy afin renommer le fichier de sauvegarde avec son nom d’origine.

Procédure
1. Dans le champ File Name de la vue Basic settings du tFileDelete, saisissez le chemin et le nom du fichier temporaire à supprimer, D:/temp/tempdata.csv dans ce scénario. Laissez les autres champs tels qu’ils sont.
2. Double-cliquez sur le **tFileCopy** afin d’ouvrir sa vue **Basic settings**.

3. Dans le champ **File Name**, saisissez le chemin et le nom du fichier de sauvegarde, *D:/temp/backup-tempdata.csv* dans ce scénario.

4. Dans le champ **Destination directory**, spécifiez le chemin ou parcourez votre système vers le dossier cible, *D:/temp/* dans ce scénario.

5. Cochez la case **Rename** et spécifiez le nom original du fichier temporaire dans le champ **Destination filename**, *tempdata.csv*.

6. Cochez la case **Remove source file** afin de supprimer le fichier de sauvegarde après l’action de renommage. Laissez les autres champs tels qu’ils sont.

**Résultat d’exécution attendu**

Si le fichier temporaire spécifié existe, ses données sont conservées dans une copie de sauvegarde avant l’exécution du Job principal et restaurées par la suite.

Si le fichier temporaire n’existe pas, les deux **tFileCopy** génèrent une erreur mais n’empêchent pas l’exécution du Job principal.

Pour le résultat de l’exécution du Job principal, consultez **Scénario 2 : Trouver des fichiers dupliqués entre deux dossiers** à la page 1160.

**Scénario associé**

Pour un autre scénario utilisant le **tPrejob**, consultez **Scénario : Créer un diagramme en barres à partir de données d’entrée** à la page 356.
tPubSubOutput

Ce composant reçoit des messages sérialisés en tableaux d'octets via le composant précédent et écrit ces messages dans un service PubSub donné.
tRedshiftBulkExec

Ce composant charge des données dans à partir d’Amazon S3, d’un cluster Amazon EMR, d’Amazon DynamoDB ou d’hôtes distants.

Les composants tRedshiftOutputBulk et tRedshiftBulkExec sont généralement utilisés ensemble comme deux parties d’un processus pour charger des données dans Amazon Redshift à partir d’un fichier CSV/délimité dans Amazon S3. Au cours de la première étape, un fichier délimité/CSV est généré. Au cours de la seconde étape, le fichier est utilisé dans une instruction INSERT utilisée pour alimenter. Cette exécution en deux étapes est unifiée dans le composant tRedshiftOutputBulkExec. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans Amazon Redshift.

Le tRedshiftBulkExec charge des données dans une table Amazon Redshift à partir d’une table Amazon DynamoDB ou de fichiers plats situé dans un bucket Amazon S3, un cluster Amazon S3, ou un hôte distant auquel accéder à l’aide d’une connexion SSH.

Propriétés du tRedshiftBulkExec Standard

Ces propriétés sont utilisées pour configurer le tRedshiftBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tRedshiftBulkExec Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Peut être Built-In ou Repository.</td>
</tr>
<tr>
<td></td>
<td>• Built-In : propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>• Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Host</td>
<td>Saisissez l’adresse IP ou le nom de l’hôte du serveur de la base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute du serveur de la base de données.</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Saisissez le nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les données d’authentification de l’utilisateur de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>.</td>
</tr>
<tr>
<td></td>
<td>Dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Additional JDBC Parameters</td>
<td>Définissez des propriétés JDBC supplémentaires pour la connexion que vous créez. Les propriétés sont séparées par une espaceclav valeur. Par exemple, <strong>ssl=true &amp; sslfactor y=com.amazon.redshift.ssl.NonValidatingFactory</strong>. Plus signifie que la connexion sera créée en utilisant le SSL.</td>
</tr>
<tr>
<td>Table Name</td>
<td>Spécifiez le nom de la table à écrire. Notez que seule une table peut être écrite à la fois.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Sur la table définie, vous pouvez effectuer l’une des opérations suivantes :</td>
</tr>
<tr>
<td></td>
<td>• <strong>None</strong> : n’effectuer aucune opération sur la table.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Drop and create table</strong> : supprimer la table puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Create table</strong> : créer une table qui n’existe pas encore.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Create table if not exists</strong> : créer la table si nécessaire.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Drop table if exists and create</strong> : supprimer la table si elle existe déjà puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Clear table</strong> : supprimer le contenu de la table.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : Le schéma existe déjà et il est stocké dans le <strong>Repository</strong>. Ainsi, il peut être réutilisé. Voir également le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
</tbody>
</table>
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Data source type**

Sélectionnez l’emplacement des données source à charger.

- **S3**: charger des données d’un fichier dans un bucket Amazon S3.
- **EMR**: charger des données d’un cluster Amazon EMR.
- **DynamoDB**: charger des données d’une table DynamoDB existante.
- **Remote host**: charger des données d’un hôte distant ou plusieurs, comme des instances de Amazon Elastic Compute Cloud (Amazon EC2) ou d’autres ordinateurs.

Pour plus d’informations, consultez Sources de données.

**Access Key**

Spécifiez l’ID de la clé d’accès identifiant de manière unique un compte AWS. Pour plus d’informations concernant l’obtention de votre clé d’accès et clé secrète, consultez Obtention de vos clés d’accès AWS.

**Secret Key**

Spécifiez la clé secrète d’accès, constituant les informations de sécurité, ainsi que la clé d’accès.

Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.

**Bucket**

Spécifiez le nom du bucket Amazon S3, contenant les données à charger.

Ce champ est disponible uniquement lorsque l’option **S3** ou **Remote host** est sélectionnée dans la liste **Data source type**.

**Key**

Saisissez la clé d’objet pour le fichier à charger dans Amazon S3.

Ce champ est disponible uniquement lorsque l’option **S3** est sélectionnée dans la liste **Data source type**.

**Cluster id**

Spécifiez l’ID du cluster stockant les données à charger.

Ce champ est disponible uniquement lorsque l’option **EMR** est sélectionnée dans la liste **Data source type**.
| **HDFS path** | Spécifiez le chemin d’accès au fichier HDFS référençant le fichier de données.  
Ce champ est disponible uniquement lorsque l’option EMR est sélectionnée dans la liste **Data source type**. |
| --- | --- |
| **Table** | Spécifiez le nom de la table DynamoDB contenant les données à charger.  
Ce champ est disponible uniquement lorsque l’option DynamoDB est sélectionnée dans la liste **Data source type**. |
| **Read ratio** | Spécifiez le pourcentage du débit fourni pour la table DynamoDB à utiliser pour le chargement des données.  
Ce champ est disponible uniquement lorsque l’option DynamoDB est sélectionnée dans la liste **Data source type**. |
| **SSH manifest file** | Spécifiez la clé d’objet pour le fichier manifest SSH fournissant les informations utilisées pour ouvrir les connexions SSH et exécuter des commandes distantes.  
Ce champ est disponible uniquement lorsque l’option Remote host est sélectionnée dans la liste **Data source type**. |

### Advanced settings

| **File type** | Sélectionnez dans la liste le type de fichier contenant les données à charger :  
- **Delimited file or CSV** : fichier délimité/CSV.  
- **JSON** : fichier JSON.  
- **AVRO** : un fichier Avro.  
- **Fixed width** : fichier à taille fixe.  
Cette liste est disponible lorsque l’option S3, EMR, ou Remote host est sélectionnée dans la liste **Data source type**. |
| --- | --- |
| **Fields terminated by** | Saisissez le caractère utilisé pour séparer les champs.  
Ce champ est disponible uniquement lorsque l’option **Delimited file or CSV** est sélectionnée dans la liste **File type**. |
| **Enclosed by** | Sélectionnez le caractère par lequel entourer les champs.  
Cette liste est disponible uniquement lorsque l’option **Delimited file or CSV** est sélectionnée dans la liste **File type**. |
| **JSON mapping** | Spécifiez comment mapper les éléments de données dans le fichier source vers les colonnes dans la table cible d’Amazon Redshift. Les valeurs valides sont : |
- **auto** : mappe les données en faisant correspondre les clés ou noms des objets dans les paires source nom/valeur pour un fichier JSON ou les noms de champs dans le schéma Avro pour un fichier Avro vers les noms de colonnes dans la table cible. L’argument est sensible à la casse et doit être entouré de guillemets doubles.

- **s3://jsonpaths_file** : mappe les données à l’aide du fichier nommé JSONPaths. Le paramètre doit être une clé d’objet Amazon S3 entourée de guillemets doubles et faisant référence à un fichier, par exemple, `s3://mybucket/jsonpaths.txt`. Pour plus d’informations, consultez [Paramètres du format de données](#).

Ce champ est disponible uniquement lorsque l’option **JSON** ou **AVRO** est sélectionnée dans la liste **File type**.

**Fixed width mapping**

Saisissez une chaîne de caractères spécifiant un libellé et une largeur de colonne personnalisés entre guillemets doubles. Le format de la chaîne de caractères est le suivant :

```
ColumnLabel1:ColumnWidth1,ColumnLabel2:ColumnWidth2,...
```

Notez que le libellé de la colonne dans la chaîne de caractères n’a aucun lien avec le nom de la colonne de la table et peut être une chaîne de caractères ou un entier. L’ordre des paires libellé/largeur doit correspondre à l’ordre exact des colonnes de la table.

Ce champ est disponible uniquement lorsque l’option **Fixed width** est sélectionnée dans la liste **File type**.

**Compressed by**

Cochez cette case et, dans la liste affiché, sélectionnez le type de compression du fichier source.

Cette case est disponible lorsque l’option **S3**, **EMR**, ou **Remote host** est sélectionnée dans la liste **Data source type**.

**Decrypt**

Cochez cette case si le fichier est crypté à l’aide du cryptage Amazon S3 côté client. Dans le champ **Encryption key** affiché, spécifiez la clé de chiffrement utilisée pour chiffrer le fichier. Notez que seule une enveloppe de clé AES 128 bits ou AES 256 bits encodée en base64 est supportée. Pour plus d’informations, consultez [Loading Encrypted Data Files from Amazon S3](#) (en anglais).

Cette case est disponible lorsque l’option **S3**, **EMR**, ou **Remote host** est sélectionnée dans la liste **Data source type**.

**Encoding**

Sélectionnez dans la liste le type d’encodage des données à charger.

Cette liste est disponible lorsque l’option **S3**, **EMR**, ou **Remote host** est sélectionnée dans la liste **Data source type**.
### Date format

Sélectionnez l’un des éléments suivants afin de spécifier le format de date dans les données source :

- **NONE** : aucun format de date n’est spécifié.
- **PATTERN** : sélectionnez cet élément et spécifiez le format de date dans le champ affiché. Le format de date par défaut est YYYY-MM-DD.
- **AUTO** : sélectionnez cet élément si vous souhaitez que Amazon Redshift reconnaisse et convertisse automatiquement le format de date.

### Time format

Sélectionnez l’un des éléments suivants dans la liste, afin de spécifier le format de l’heure dans les données source :

- **NONE** : aucun format d’heure n’est spécifié.
- **PATTERN** : sélectionnez cet élément et spécifiez le format de l’heure dans le champ affiché. Le format de l’heure par défaut est YYYY-MM-DD HH:MI:SS.
- **AUTO** : sélectionnez cet élément si vous souhaitez qu’Amazon Redshift reconnaisse et convertisse le format de l’heure.
- **EPOCHSECS** : sélectionnez cet élément si les données source sont représentées comme temps epoch, soit le nombre de secondes depuis Jan 1, 1970 00:00:00 UTC.
- **EPOCHMILLISECS** : sélectionnez cet élément si les données source sont représentées comme temps epoch, soit le nombre de millisecondes depuis Jan 1, 1970 00:00:00 UTC.

### Settings

Cliquez sur le bouton [+] sous la table afin de spécifier plus de paramètres pour le chargement des données.

- **Parameter** : cliquez sur la cellule et sélectionnez un paramètre dans la liste déroulante.
- **Value** : configurez la valeur du paramètre correspondant. Notez que vous ne pouvez configurer la valeur d’un paramètre (par exemple **IGNOREBLANKLINES**) ne nécessitant pas de valeur.


### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

## Utilisation

### Règle d’utilisation

Le composant **tRedshiftBulkExec** supporte le chargement des données dans Amazon Redshift depuis un fichier délimité/CSV, JSON, ou à taille fixe, dans Amazon S3, mais le composant **tRedshiftOutputBulk** supporte uniquement la génération et le chargement d’un fichier délimité/CSV dans Amazon S3. Lorsque vous devez charger des données depuis un fichier JSON ou à taille fixe, vous pouvez utiliser le composant **tFileOutputJSON** ou **tFileOutputPositional** avec le **tS3Put** au lieu d’utiliser le composant **tRedshiftOutputBulk** pour générer et charger le fichier dans Amazon S3.

### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez [Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte](#) à la page 2641 et [Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement](#) à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.  

---

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Utilisation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tRedshiftBulkExec</strong> supporte le chargement des données dans Amazon Redshift depuis un fichier délimité/CSV, JSON, ou à taille fixe, dans Amazon S3, mais le composant <strong>tRedshiftOutputBulk</strong> supporte uniquement la génération et le chargement d’un fichier délimité/CSV dans Amazon S3. Lorsque vous devez charger des données depuis un fichier JSON ou à taille fixe, vous pouvez utiliser le composant <strong>tFileOutputJSON</strong> ou <strong>tFileOutputPositional</strong> avec le <strong>tS3Put</strong> au lieu d’utiliser le composant <strong>tRedshiftOutputBulk</strong> pour générer et charger le fichier dans Amazon S3.</td>
<td></td>
</tr>
</tbody>
</table>

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez [Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte](#) à la page 2641 et [Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement](#) à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*. |
Scénario : Charger/retirer des données dans/de Amazon S3

Ce scénario décrit un Job générant un fichier délimité, le chargeant le fichier dans S3, chargeant des données du fichier de S3 dans Redshift et les affichant dans la console. Ce Job retire ensuite les données de Redshift, les ajoute dans des fichiers S3 pour chaque slice du cluster Redshift puis liste et obtient les fichiers retirés dans S3.

Prérequis :

Les variables de contexte sont créées et sauvegardées dans la vue Repository. Pour plus d'informations concernant les variables de contexte, consultez le Guide utilisateur du Studio Talend.

- redshift_host : URL de l'endpoint de connexion au cluster Redshift.
- redshift_port : numéro du port d'écoute du serveur de la base de données.
- redshift_database : nom de la base de données.
- redshift_username : nom de l’utilisateur pour l’authentification à la base de données.
- redshift_password : mot de passe pour l’authentification à la base de données.
• **redshift_schema** : nom du schéma.
• **s3_accesskey** : clé d’accès à Amazon S3.
• **s3_secretkey** : clé secrète pour accéder à Amazon S3.
• **s3_bucket** : nom du bucket Amazon S3.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type</td>
<td>Default</td>
<td>Value</td>
</tr>
<tr>
<td>redshift_host</td>
<td>String</td>
<td></td>
<td>redshift-cluster.abcdeghi01.us-east-1.redshift.amazonaws.com</td>
</tr>
<tr>
<td>redshift_port</td>
<td>String</td>
<td></td>
<td>5439</td>
</tr>
<tr>
<td>redshift_database</td>
<td>String</td>
<td></td>
<td>redshiftdb</td>
</tr>
<tr>
<td>redshift_username</td>
<td>String</td>
<td></td>
<td>talend</td>
</tr>
<tr>
<td>redshift_password</td>
<td>Password</td>
<td></td>
<td>************</td>
</tr>
<tr>
<td>s3_accesskey</td>
<td>String</td>
<td></td>
<td>ABCDEFGHIJKLMNOPQRSTUVWXYZ</td>
</tr>
<tr>
<td>s3_secretkey</td>
<td>String</td>
<td></td>
<td>aB1cDef2gHjk+12mNoPQrsT+h5+6ypvDGTSRYU4</td>
</tr>
<tr>
<td>s3_bucket</td>
<td>String</td>
<td></td>
<td>redshift-bucket-talend</td>
</tr>
</tbody>
</table>

Notez que toutes les valeurs de contexte dans la capture d’écran ci-dessus ont pour seul objectif la démonstration.

**Ajouter et relier les composants**

**Procédure**
2. Ajoutez les composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette : un composant tRowGenerator, un tRedshiftOutputBulk, un tRedshiftBulkExec, un tRedshiftInput, un tLogRow, un tRedshiftUnload, un tS3List et un tS3Get.
3. Reliez le tRowGenerator au tRedshiftOutputBulk à l’aide d’un lien Row > Main.
4. Répétez l’opération afin de relier le tRedshiftInput au tLogRow.
5. Reliez le tS3List au tS3Get à l’aide d’un lien Row > Iterate.
6. Reliez le tRowGenerator au tRedshiftBulkExec à l’aide d’un lien Trigger > On Subjob Ok.
7. Répétez l’opération pour relier le tRedshiftBulkExec au tRedshiftInput, le tRedshiftInput au tRedshiftUnload et le tRedshiftUnload au tS3List.

**Configurer les composants**

**Préparer un fichier et le charger dans S3**

**Procédure**
1. Double-cliquez sur le tRowGenerator afin d’ouvrir son éditeur RowGenerator Editor.
2. Cliquez deux fois sur le bouton [+] afin d’ajouter deux colonnes : ID, de type Integer et Name, de type String.

3. Cliquez dans la cellule de la colonne Functions et sélectionnez dans la liste une fonction pour chaque colonne. Dans cet exemple, sélectionnez Numeric.sequence pour générer des nombres en séquence pour la colonne ID et sélectionnez TalendDataGenerator.getFirstName afin de générer des prénoms aléatoires pour la colonne Name.

4. Dans le champ Number of Rows for RowGenerator, saisissez le nombre de lignes de données à générer. Dans cet exemple, saisissez 20.

5. Cliquez sur OK pour fermer l’éditeur de schéma et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

6. Double-cliquez sur le tRedshiftOutputBulk pour ouvrir sa vue Basic settings.

7. Dans le champ Data file path at local, spécifiez le chemin local pour le fichier à générer. Dans cet exemple, le chemin est E:/Redshift/redshift_bulk.txt.

8. Dans le champ Access Key, appuyez sur les touches Ctrl + Espace et, dans la liste, sélectionnez context.s3_accesskey afin de renseigner ce champ.

   Répétez l’opération dans le champ Secret Key avec la valeur context.s3_secretkey et dans le champ Bucket avec la valeur context.s3_bucket.

9. Dans le champ Key, saisissez un nouveau nom pour le fichier à générer après chargement dans Amazon S3. Dans cet exemple, saisissez person_load.
Charger les données du fichier de S3 dans Redshift

Procédure

1. Double-cliquez sur le tRedshiftBulkExec pour ouvrir sa vue Basic settings.

2. Dans le champ Host, appuyez sur les touches Ctrl + Espace et, dans la liste, sélectionnez context.redshift_host afin de renseigner ce champ.

Répétez l’opération pour :

- le champ Port avec la valeur context.redshift_port,
- le champ Database avec la valeur context.redshift_database,
- le champ Schema avec la valeur context.redshift_schema,
- le champ Username avec la valeur context.redshift_username,
- le champ Password avec la valeur context.redshift_password,
- le champ Access Key avec la valeur context.s3_accesskey,
- le champ Secret Key avec la valeur context.s3_secretkey et
- le champ Bucket avec la valeur context.s3_bucket.

3. Dans le champ Table Name, saisissez le nom de la table à écrire. Dans cet exemple, saisissez person.

4. Dans la liste Action on table, sélectionnez Drop table if exists and create.

5. Dans le champ Key, saisissez le nom du fichier de Amazon S3 à charger. Dans cet exemple, saisissez person_load.

6. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la fenêtre qui s’ouvre, définissez le schéma en ajoutant deux colonnes : ID de type Integer et Name, de type String.
Récupérer les données de la table dans Redshift

**Procédure**

1. Double-cliquez sur le composant **tRedshiftInput** pour ouvrir sa vue **Basic settings**.

2. Renseignez les champs **Host**, **Port**, **Database**, **Schema**, **Username** et **Password** avec les variables de contexte correspondantes.

3. Dans le champ **Table Name**, saisissez le nom de la table à lire. Dans cet exemple, saisissez **person**.

4. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans la fenêtre qui s’ouvre, définissez le schéma en ajoutant deux colonnes : **ID** de type **Integer** et **Name**, de type **String**.

5. Dans le champ **Query**, saisissez l'instruction SQL suivante permettant de récupérer les données.

   ```sql
   "SELECT * FROM " + context.redshift_schema + "person ORDER BY "ID"
   ```

6. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.
7. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

**Retirer des données de Redshift et les ajouter dans des fichiers S3**

**Procédure**

1. Double-cliquez sur le **tRedshiftUnload** pour ouvrir sa vue **Basic settings**.

2. Renseignez les champs **Host**, **Port**, **Database**, **Schema**, **Username** et **Password** avec les variables de contexte correspondantes.
   Renseignez les champs **Access Key**, **Secret Key** et **Bucket** avec les variables de contexte correspondantes.

3. Dans le champ **Table Name**, saisissez le nom de la table de laquelle lire les données. Dans cet exemple, saisissez *person*.

4. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans la fenêtre qui s'ouvre, définissez le schéma en ajoutant deux colonnes : *ID*, de type **Integer** et *Name*, de type **String**.

5. Dans le champ **Query**, saisissez l'instruction SQL suivante à partir de laquelle les résultats seront retirés.

"SELECT * FROM person"
6. Dans le champ **Key prefix**, saisissez le préfixe du nom pour les fichiers à retirer. Dans cet exemple, saisissez `person_unload_`.

**Récupérer les fichiers retirés de Amazon S3**

**Procédure**

1. Double-cliquez sur le composant **tS3List** pour ouvrir sa vue **Basic settings**.

![Image du composant tS3List](image1.png)

2. Renseignez les champs **Access Key** et **Secret Key** avec les variables de contexte correspondantes.
4. Décrochez la case **List all buckets objects** et cliquez sur le bouton `+` sous la table pour ajouter une ligne.
   - Renseignez la colonne **Bucket name** avec le nom du bucket dans lequel les fichiers retirés sont créés. Dans cet exemple, saisissez la variable de contexte `context.s3_bucket`.
   - Renseignez la colonne **Key prefix** avec le préfixe du nom des fichiers retirés. Dans cet exemple, saisissez `person_unload_`.
5. Double-cliquez sur le **tS3Get** pour ouvrir sa vue **Basic settings**.

![Image du composant tS3Get](image2.png)

6. Renseignez les champs **Access Key** et **Secret Key** avec les variables de contexte correspondantes.

8. Dans le champ Bucket, saisissez le nom du bucket dans lequel les fichiers retirés sont créés. Dans cet exemple, saisissez la variable de contexte context.s3_bucket.

Dans le champ Key, saisissez le nom des fichiers retirés, en appuyant sur les touches Ctrl+Espace et, dans la liste, sélectionnez la variable globale ((String)globalMap.get("tS3List_1_CURRENT_KEY")).

9. Dans le champ File, saisissez le chemin d'accès local à l'emplacement où les fichiers retirés sont sauvegardés. Dans cet exemple, saisissez "E:/Redshift/ + ((String)globalMap.get("tS3List_1_CURRENT_KEY"))

Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur les touches Ctrl + S afin de sauvegarder votre Job.
2. Exécutez le Job en appuyant sur F6 ou en appuyant sur le bouton Run de la vue Run.

[statistics] connecting to socket on port 3726
[statistics] connected

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Abraham</td>
</tr>
<tr>
<td>2</td>
<td>Herbert</td>
</tr>
<tr>
<td>3</td>
<td>Chester</td>
</tr>
<tr>
<td>4</td>
<td>John</td>
</tr>
<tr>
<td>5</td>
<td>Gerald</td>
</tr>
<tr>
<td>6</td>
<td>Lyndon</td>
</tr>
<tr>
<td>7</td>
<td>Warren</td>
</tr>
<tr>
<td>8</td>
<td>Calvin</td>
</tr>
<tr>
<td>9</td>
<td>Chester</td>
</tr>
<tr>
<td>10</td>
<td>Herbert</td>
</tr>
<tr>
<td>11</td>
<td>Ronald</td>
</tr>
<tr>
<td>12</td>
<td>Herbert</td>
</tr>
<tr>
<td>13</td>
<td>Martin</td>
</tr>
<tr>
<td>14</td>
<td>James</td>
</tr>
<tr>
<td>15</td>
<td>Franklin</td>
</tr>
<tr>
<td>16</td>
<td>Herbert</td>
</tr>
<tr>
<td>17</td>
<td>Richard</td>
</tr>
<tr>
<td>18</td>
<td>Warren</td>
</tr>
<tr>
<td>19</td>
<td>Jimmy</td>
</tr>
<tr>
<td>20</td>
<td>George</td>
</tr>
</tbody>
</table>

[statistics] disconnected
Comme affiché ci-dessus, les données générées sont écrites dans le fichier local redshift_bulk.txt. Le fichier est chargé dans S3 avec un nouveau nom, person_load. Les données sont chargées du fichier S3 dans la table person de Redshift et affichées dans la console. Les données sont ensuite retirées de la table person de Redshift et ajoutées à deux fichiers person_unload_0000_part_00 et person_unload_0001_part_00 dans S3, par slice du cluster Redshift, puis les fichiers retirés de S3 sont listés et récupérés dans le dossier local.
tRedshiftClose

Ce composant assure l’intégrité de la transaction effectuée dans la base de données.
Ce composant est utilisé avec le tRedShiftConnection et le tRedshiftCommit
Le composant tRedshiftClose ferme la connexion à une base de données connectée.

Propriétés du tRedshiftClose Standard

Ces propriétés sont utilisées pour configurer le tRedshiftClose s’exécutant dans le framework de Jobs Standard.
Le composant tRedshiftClose Standard appartient aux familles Cloud et Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tRedshiftConnection dans la liste.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec les composants Amazon Redshift, notamment le tRedshiftConnection et le tRedshiftCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez...</td>
</tr>
</tbody>
</table>
pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Pour un scénario associé, consultez Scénario : Manipuler des données avec Redshift à la page 3323.
tRedshiftCommit

Ce composant améliore les performances.
En utilisant une connexion unique, commitez en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes.

Le composant tRedshiftCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tRedshiftCommit Standard

Ces propriétés sont utilisées pour configurer le tRedshiftCommit s’exécutant dans le framework de Jobs Standard.

Le composant tRedshiftCommit Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tRedshiftConnection dans la liste.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Avertissement :
Si vous utilisez un lien de type Row >Main pour relier le tRedshiftCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.

Advanced settings

| tStat Catcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants Amazon Redshift, notamment le tRedshiftConnection et le tRedshiftRollback.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Scénario associé

Pour un scénario du même type, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tRedshiftConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Le composant tRedshiftConnection ouvre une connexion vers une base de données afin d’effectuer une transaction.

**Propriétés du tRedshiftConnection Standard**

Ces propriétés sont utilisées pour configurer le tRedshiftConnection s’exécutant dans le framework de Jobs Standard.

Le composant tRedshiftConnection Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
Additional JDBC Parameters

Définissez des propriétés JDBC supplémentaires pour la connexion que vous créez. Les propriétés sont séparées par une espace et chaque propriété est une paire clé-valeur. Par exemple, `ssl=true & sslfactor=com.amazon.redshift.ssl.NonValidatingFactory`, qui signifie que la connexion sera créée en utilisant le SSL.

Use or register a shared DB Connection

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ Shared DB Connection Name qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options Use dynamic job et Use an independent process to run subjob du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.

Advanced settings

Auto Commit

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne committe pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

Règle d’utilisation

Ce composant est généralement utilisé avec des composants Amazon Redshift, notamment les composants tRedshiftCommit et tRedshiftRollback.

Utilisation
<table>
<thead>
<tr>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez **Scénario : Manipuler des données avec Redshift** à la page 3323.
tRedshiftInput

Ce composant lit une base de données et en extraite des champs à l’aide de requêtes.

Le composant tRedshiftInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

Propriétés du tRedshiftInput Standard

Ces propriétés sont utilisées pour configurer le tRedshiftInput s’exécutant dans le framework de Jobs Standard.

Le composant tRedshiftInput Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td></td>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :
Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Additional JDBC Parameters</td>
<td>Définissez des propriétés JDBC supplémentaires pour la connexion que vous créez. Les propriétés sont séparées par une espace et chaque propriété est une paire clé-valeur. Par exemple, sssl=true &amp; sslfactor=y=com.amazon.redshift.ssl.NonValidatingFactory, qui signifie que la connexion sera créée en utilisant le SSL.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td>Built-In</td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Repository</td>
<td>Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
<td></td>
</tr>
</tbody>
</table>
- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **Repository Content**.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Nom de la table de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query Type et Query</strong></td>
<td>Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.</td>
</tr>
<tr>
<td><strong>Guess Query</strong></td>
<td>Cliquez sur le bouton <strong>Guess Query</strong> pour générer la requête correspondant au schéma de votre table dans le champ <strong>Query</strong>.</td>
</tr>
<tr>
<td><strong>Guess schema</strong></td>
<td>Cliquez sur ce bouton pour récupérer le schéma de la table.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| Log file | Spécifiez le chemin du fichier de log à générer pendant le processus de récupération des données.
Notez que si les informations de log doivent être sauvegardées et que si ce composant réutilise la connexion créée par un composant tRedshiftConnection, vous devez spécifier la propriété loglevel=1/2/3 dans le champ **Additional JDBC Parameters** du composant tRedshiftConnection. |
| Logging level | Dans la liste déroulante, sélectionnez un niveau de log pour définir le type d’événements à enregistrer dans le fichier de log. |
| Use cursor | Cochez cette case et définissez le nombre de lignes avec lesquelles vous souhaitez travailler en une fois. Cette option permet d’optimiser les performances. |
| Trim all the String/Char columns | Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaines de caractères. |
| Trim column | Supprime les espaces en début et en fin de champ dans les colonnes sélectionnées. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
Global Variables

Global Variables

NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Amazon Redshift.

Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

**Limitation**

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

---

**Scénario : Manipuler des données avec Redshift**

Ce scénario décrit un Job qui écrit des informations personnelles dans Redshift puis récupère ces informations et les affiche dans la console.

Le scénario utilise les six composants suivants :

- un **tRedshiftConnection**, qui ouvre une connexion à Redshift.
- un **tFixedFlowInput**, qui définit la structure des informations personnelles et envoie ces données au composant suivant.
- un **tRedshiftOutput**, qui écrit les données reçues par le composant précédent dans Redshift.
- un **tRedshiftInput**, qui lit les données depuis Redshift.
- un **tLogRow**, qui affiche les données reçues par le composant précédent dans la console.
- et un **tRedshiftClose**, qui ferme la connexion à Redshift.
Déposer et relier les composants

**Procédure**

1. Déposez les six composants listés précédemment depuis la Palette dans l'espace de modélisation graphique.
2. Reliez le tFixedFlowInput au tRedshiftOutput à l'aide d'un lien Row > Main.
3. Reliez le tRedshiftInput au tLogRow à l'aide d'un lien Row > Main.
4. Reliez le tRedshiftConnection au tFixedFlowInput à l'aide d'un lien Trigger > OnSubjobOk.
5. Reliez le tFixedFlowInput au tRedshiftInput et le tRedshiftInput au tRedshiftClose à l'aide de liens Trigger > OnSubjobOk.

Configurer les composants

**Ouvrir une connexion à Redshift**

**Procédure**

1. Double-cliquez sur le tRedshiftConnection afin d'ouvrir sa vue Basic settings.

2. Sélectionnez Built-In depuis la liste Property Type.
   - Dans les champs Host, Port, Database, Schema, Username et Password, saisissez les informations nécessaires pour se connecter à Redshift.
3. Dans la vue Advanced settings, cochez la case Auto Commit afin de commiter tous les changements à vers Redshift à chaque transaction.

Définir les données d’entrée

**Procédure**

1. Double-cliquez sur le tFixedFlowInput afin d’ouvrir sa vue Basic settings.
2. Cliquez sur le bouton [...] à côté du champ **Edit schema** afin d’ouvrir l’éditeur de schéma.

3. Dans l’éditeur de schéma, cliquez sur le bouton 

4. Cliquez sur **OK** pour valider les modifications puis acceptez la propagation proposée par la fenêtre pop-up **[Propagate]** en appuyant sur le bouton **Yes**.

5. Dans la zone **Mode**, sélectionnez **Use Inline Content (delimited file)** et saisissez les informations personnelles dans le champ **Content**.

Écrire les données dans Redshift

Procédure

1. Double-cliquez sur le composant **tRedshiftOutput** pour ouvrir sa vue **Basic settings**.
Cochez la case **Use an existing connection** puis sélectionnez la connexion précédemment configurée dans le **tRedshiftConnection**, dans la liste déroulante **Component List**.

Dans le champ **Table**, parcouriez ou saisissez le chemin d'accès à la table dans laquelle écrire les données, *redshiftexample* dans ce scénario.

Sélectionnez **Drop table if exists and create** dans la liste **Action on table** et sélectionnez **Insert** dans la liste **Action on data**.

Cliquez sur **Sync columns** afin de récupérer le schéma du composant précédent.

Récupérer les données de Redshift

**Procédure**

1. Double-cliquez sur le **tRedshiftInput** pour ouvrir sa vue **Basic settings**.

2. Cochez la case **Use an existing connection** et sélectionnez la connexion précédemment configurée dans le **tRedshiftConnection**, dans la liste **Component List**.

3. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l'éditeur de schéma.

4. Dans l'éditeur de schéma, cliquez trois fois sur le bouton [+] pour ajouter trois colonnes, respectivement : *id*, de type **Integer**, *name*, de type **String** et *age*, de type **Integer**. La structure des données est la même que celle définie dans le **tFixedFlowInput**.

5. Cliquez sur **OK** afin de valider les modifications, puis acceptez la propagation proposée par la fenêtre **[Propagate]** qui s'ouvre.

6. Dans le champ **Table Name**, parcouriez ou saisissez le chemin d'accès à la table dans laquelle vous souhaitez écrire les données, *redshiftexample* dans cet exemple.

7. Cliquez sur le bouton **Guess Query** pour générer la requête. Le champ **Query** contient automatiquement la requête générée.
Afficher les informations définies

Procédure
1. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

2. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un meilleur affichage des résultats.

Fermer la connexion à Redshift

Procédure
1. Double-cliquez sur le **tRedshiftClose** pour ouvrir sa vue **Basic settings**.

2. Dans la liste **Component List**, sélectionnez la connexion précédemment définie dans le **tRedshiftConnection**.

Sauvegarder et exécuter le Job

Procédure
1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder le Job.
2. Appuyez sur **F6** pour l’exécuter.
Les informations personnelles sont écrites dans la base de données cible Redshift et les données sont récupérées de la base de données et affichées dans la console.
tRedshiftOutput

Ce composant écrit, met à jour, modifie ou supprime les données d’une base de données.

Le composant tRedshiftOutput exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Propriétés du tRedshiftOutput Standard

Ces propriétés sont utilisées pour configurer le tRedshiftOutput s’exécutant dans le framework de Jobs Standard.

Le composant tRedshiftOutput Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Nom exact du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Additional JDBC Parameters</td>
<td>Définissez des propriétés JDBC supplémentaires pour la connexion que vous créez. Les propriétés sont séparées par une espace et chaque propriété est une paire clé-valeur. Par exemple, ssl=true &amp; ssifactor y=com.amazon.redshift.ssl.NonValidatingFactory, qui signifie que la connexion sera créée en utilisant le SSL.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
  - **None** : n’effectuer aucune opération de table.  
  - **Drop and create the table** : supprimer la table puis en créer une nouvelle.  
  - **Create a table** : créer une table qui n’existe pas encore.  
  - **Create table if doesn’t exist** : créer la table si nécessaire.  
  - **Drop a table if exists and create** : supprimer la table si elle existe déjà, puis en créer une nouvelle.  
  - **Clear a table** : supprimer le contenu de la table. |
| Action on data | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : |
**Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.

**Update** : Mettre à jour les entrées existantes.

**Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.

**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

**Delete** : Supprimer les entrées correspondantes au flux d’entrée.

⚠️ **Avertissement** :
Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

---

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

**Repository** : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

**Extend Insert**

Cochez cette case pour effectuer une insertion de masse d’un ensemble de lignes défini et non les insérer ligne par ligne. Le gain de performance est considérable.

**Number of rows per insert** : saisissez le nombre de lignes à insérer par opération. Notez que, plus la valeur spécifiée est grande, plus la performance est faible, à cause de l’augmentation de la sollicitation de la mémoire.

Amazon Redshift demande un nombre de lignes par insertion inférieur à 32 767. Si le nombre que vous saisissez dépasse cette limite, le Studio modifie ce nombre pour qu’il soit juste sous la limite.

*i Remarque :*

Cette option est incompatible avec le lien **Reject**. Vous devez décochez la case si vous utilisez un lien **Row > Rejects** avec ce composant.

**Use Batch**

Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.

*i Remarque :*

Cette case est disponible uniquement lorsque vous sélectionnez **Update** ou **Delete** dans la liste **Action on data**.

**Batch Size**

Spécifiez le nombre d’enregistrements à traiter dans chaque lot.
Ce champ est disponible uniquement lorsque la case **Use batch mode** est cochée.

**Commit every**

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

**Additional Columns**

Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.

**Name** : Saisissez le nom de la colonne à modifier ou à insérer.

**SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.

**Position** : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l’action à effectuer sur la colonne de référence.

**Reference column** : Saisissez une colonne de référence que le composant **tDBOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

**Use field options**

Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

#### Règle d’utilisation

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités des requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données Amazon Redshift. Il permet aussi de créer un flux de rejet avec un lien **Row > Reject** filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien **Reject** à la page 2675.

#### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.


#### Limitation

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les...

Scénarios associés

Pour un scénario associé, consultez Scénario : Manipuler des données avec Redshift à la page 3323.
tRedshiftOutputBulk

Ce composant prépare un fichier délimité/CSV pouvant être utilisé par le tRedshiftBulkExec pour alimenter Amazon Redshift.

Les composants tRedshiftOutputBulk et tRedshiftBulkExec sont généralement utilisés ensemble comme deux parties d’un processus pour charger des données dans Amazon Redshift à partir d’un fichier CSV/délimité dans Amazon S3. Au cours de la première étape, un fichier délimité/CSV est généré. Au cours de la seconde étape, le fichier est utilisé dans une instruction INSERT utilisée pour alimenter. Cette exécution en deux étapes est unifiée dans le composant tRedshiftOutputBulkExec. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans Amazon Redshift.

Le composant tRedshiftOutputBulk reçoit des données du composant précédent, génère un fichier délimité/CSV et charge le fichier dans Amazon S3.

Propriétés du tRedshiftOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tRedshiftOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tRedshiftOutputBulk Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data file path at local</td>
<td>Spécifiez le chemin d’accès local au fichier à générer. Notez que ce fichier est généré sur la machine où est installé le Studio ou sur laquelle le Job utilisant ce composant est déployé.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
</tbody>
</table>

Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.
**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th><strong>Append the local file</strong></th>
<th>Cochez cette case pour écrire à la suite les données dans le fichier local spécifié s’il existe déjà, au lieu de l’écraser.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Compress the data file</strong></th>
<th>Cochez cette case et sélectionnez dans la liste un type de compression pour le fichier de données. Cette case est indisponible lorsque la case <strong>Append the local file</strong> est cochée.</th>
</tr>
</thead>
</table>

| **Encrypt** | Cochez cette case pour générer et charger le fichier de données dans Amazon S3 en utilisant le chiffrement côté client. Dans le champ **Encryption key** affiché, saisissez la clé de chiffrement utilisée pour chiffrer le fichier.  
Cette case est décochée par défaut et le fichier de données est chargé dans Amazon S3 en utilisant le chiffrement côté serveur.  
Pour plus d’informations sur les chiffrements côté client et côté serveur, consultez Protecting Data Using Encryption. |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------|

<table>
<thead>
<tr>
<th><strong>Access Key</strong></th>
<th>Spécifiez l’ID de la clé d’accès identifiant de manière unique un compte AWS. Pour plus d’informations concernant l’obtention de votre clé d’accès et clé secrète, consultez Obtention de vos clés d’accès AWS.</th>
</tr>
</thead>
</table>

| **Secret Key** | Spécifiez la clé secrète d’accès, constituant les informations de sécurité, ainsi que la clé d’accès.  
Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|
**Bucket**
Saisissez le nom du bucket Amazon S3, le dossier de plus haut niveau, dans lequel charger le fichier.

**Key**
Saisissez la clé d'objet assignée au fichier chargé dans Amazon S3.

### Advanced settings

<table>
<thead>
<tr>
<th>Field Separator</th>
<th>Saisissez le caractère utilisé pour séparer les champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text enclosure</td>
<td>Sélectionnez le caractère avec lequel entourer les champs.</td>
</tr>
<tr>
<td>Delete local file after putting it to s3</td>
<td>Cochez cette case pour supprimer le fichier local après son chargement dans Amazon S3. Par défaut, cette case est cochée.</td>
</tr>
<tr>
<td>Create directory if not exists</td>
<td>Cochez cette case pour créer le répertoire spécifié dans le champ <strong>Data file path at local</strong> s'il n'existe pas. Par défaut, cette case est cochée.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Sélectionnez un type d'encodage pour les données dans le fichier à générer.</td>
</tr>
</tbody>
</table>
| Config client | Cochez cette case afin de configurer les paramètres clients pour Amazon S3. Cliquez sur le bouton [+] sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre client et configurez les attributs suivants pour chaque paramètre :
- **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.
- **Value** : saisissez la valeur du paramètre client correspondant. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant. |

### Global Variables

**Global Variables**

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable *After* et retourne un entier.

**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l'exécution d'un composant. Une variable *After* fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec le composant **tRedshiftBulkExec** pour alimenter Amazon Redshift avec un fichier délimité/CSV. Utilisés ensemble, ils permettent de maximiser les performances lorsqu’ils alimentent Amazon Redshift. |

**Scénario associé**

Pour un scénario associé, consultez **Scénario : Charger/retirer des données dans/de Amazon S3** à la page 3303.
tRedshiftOutputBulkExec

Ce composant exécute l’action Insert sur les données fournies.

En tant que composant dédié, ce composant permet un gain de performance pendant les opérations d’Insert dans Amazon Redshift.

Les composants tRedshiftOutputBulk et tRedshiftBulkExec sont généralement utilisés ensemble comme deux parties d’un processus pour charger des données dans Amazon Redshift à partir d’un fichier CSV/délimité dans Amazon S3. Au cours de la première étape, un fichier délimité/CSV est généré. Au cours de la seconde étape, le fichier est utilisé dans une instruction INSERT utilisée pour alimenter. Cette exécution en deux étapes est unifiée dans le composant tRedshiftOutputBulkExec.

L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le changement des données dans Amazon Redshift.

Le composant tRedshiftOutputBulkExec reçoit des données du composant précédent, génère un fichier délimité/CSV, charge le fichier dans Amazon S3, puis charge les données d’Amazon S3 dans Redshift.

Propriétés du tRedshiftOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tRedshiftOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tRedshiftOutputBulkExec Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Peut être Built-In ou Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Saisissez l'adresse IP ou le nom de l'hôte du serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Saisissez le numéro du port d'écoute du serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Saisissez le nom du schéma.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez les données d'authentification de l'utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Additional JDBC Parameters</strong></td>
<td>Définissez des propriétés JDBC supplémentaires pour la connexion que vous créez. Les propriétés sont séparées par une esperluette et chaque propriété est une paire clé-valeur. Par exemple, ssl=true &amp; ssifactor y=com.amazon.redshift.ssl.NonValidatingFactory, qui signifie que la connexion sera créée en utilisant le SSL.</td>
</tr>
<tr>
<td><strong>Table Name</strong></td>
<td>Spécifiez le nom de la table à écrire. Notez que seule une table peut être écrête à la fois.</td>
</tr>
<tr>
<td><strong>Action on table</strong></td>
<td>Sur la table définie, vous pouvez effectuer l'une des opérations suivantes :</td>
</tr>
<tr>
<td></td>
<td>• None : n'effectuer aucune opération sur la table.</td>
</tr>
<tr>
<td></td>
<td>• Drop and create table : supprimer la table puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td>• Create table : créer une table qui n'existe pas encore.</td>
</tr>
<tr>
<td></td>
<td>• Create table if not exists : créer la table si nécessaire.</td>
</tr>
<tr>
<td></td>
<td>• Drop table if exists and create : supprimer la table si elle existe déjà, puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td>• Clear table : supprimer le contenu de la table.</td>
</tr>
<tr>
<td><strong>Schema et Edit schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Data file path at local**

Spécifiez le chemin d'accès local du fichier à générer. Notez que le fichier est généré sur la machine où est installé le Studio ou sur celle où le Job utilisant ce composant est déployé.

**Append the local file**

Cochez cette case pour écrire à la suite les données dans le fichier local spécifié s'il existe déjà, au lieu de l'éraser.

**Create directory if not exists**

Cochez cette case pour créer le répertoire spécifié dans le champ **Data file path at local** s'il n'existe pas. Par défaut, cette case est cochée.

**Access Key**

Spécifiez l'ID de la clé d'accès identifiant de manière unique un compte AWS. Pour plus d'informations concernant l'obtention de votre clé d'accès et clé secrète, consultez **Obtention de vos clés d'accès AWS**.

**Secret Key**

Spécifiez la clé secrète d'accès, constituant les informations de sécurité, ainsi que la clé d'accès. Pour saisir la clé secrète, cliquez sur le bouton […] à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.

**Bucket**

Saisissez le nom du bucket Amazon S3, le dossier de plus haut niveau, dans lequel charger le fichier.

**Key**

Saisissez la clé d'objet assignée au fichier chargé dans Amazon S3.

**Advanced settings**

**Fields terminated by**

Saisissez le caractère utilisé pour séparer les champs.
<table>
<thead>
<tr>
<th>Enclosed by</th>
<th>Sélectionnez le caractère avec lequel entourer les champs.</th>
</tr>
</thead>
</table>
| Compressed by     | Cochez cette case et sélectionnez dans la liste un type de compression pour le fichier de données.  
Cette case est indisponible lorsque la case **Append the local file** est cochée. |
| Encrypt           | Cochez cette case pour générer et charger le fichier de données à l’aide du cryptage Amazon S3 côté client. 
Dans le champ **Encryption key** affiché, spécifiez la clé de chiffrement utilisée pour chiffrer le fichier. Notez que seule une enveloppe de clé AES 128 bits ou AES 256 bits encodée en base64 est supportée. Pour plus d’informations, consultez Loading Encrypted Data Files from Amazon S3 (en anglais). 
Cette case est décochée par défaut et le fichier de données est chargé dans Amazon S3 en utilisant le chiffrement côté serveur. 
Pour plus d’informations sur les chiffrements côté client et côté serveur, consultez Protecting Data Using Encryption. |
| Encoding          | Sélectionnez un type d’encodage pour les données dans le fichier à généérer. |
| Delete local file after putting it to s3 | Cochez cette case pour supprimer le fichier local après son chargement dans Amazon S3. Par défaut, cette case est cochée. |
| Date format       | Sélectionnez l’un des éléments suivants afin de spécifier le format de date dans les données source :  
• **NONE** : aucun format de date n’est spécifié.  
• **PATTERN** : sélectionnez cet élément et spécifiez le format de date dans le champ affiché. Le format de date par défaut est YYYY-MM-DD.  
• **AUTO** : sélectionnez cet élément si vous souhaitez que Amazon Redshift reconnaisse et convertisse automatiquement le format de date. |
| Time format       | Sélectionnez l’un des éléments suivants dans la liste, afin de spécifier le format de l’heure dans les données source :  
• **NONE** : aucun format d’heure n’est spécifié.  
• **PATTERN** : sélectionnez cet élément et spécifiez le format de l’heure dans le champ affiché. Le format de l’heure par défaut est YYYY-MM-DD HH:MI:SS.  
• **AUTO** : sélectionnez cet élément si vous souhaitez qu’Amazon Redshift reconnaisse et convertisse le format de l’heure.  
• **EPOCHSECS** : sélectionnez cet élément si les données source sont représentées comme temps |
epoch, soit le nombre de secondes depuis Jan 1, 1970 00:00:00 UTC.

- **EPOCHMILLISECS** : sélectionnez cet élément si les données source sont représentées comme temps epoch, soit le nombre de millisecondes depuis Jan 1, 1970 00:00:00 UTC.

### Settings

Cliquez sur le bouton [+] sous la table afin de spécifier plus de paramètres pour le chargement des données.

- **Parameter** : cliquez sur la cellule et sélectionnez un paramètre dans la liste déroulante.
- **Value** : configurez la valeur du paramètre correspondant. Notez que vous ne pouvez configurer la valeur d’un paramètre (par exemple **IGNOREBLANKLINES**) ne nécessitant pas de valeur.


### Config client

Cochez cette case afin de configurer les paramètres clients pour Amazon S3. Cliquez sur le bouton [+] sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre client et configurez les attributs suivants pour chaque paramètre :

- **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.
- **Value** : saisissez la valeur du paramètre client correspondant.

### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option. Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</th>
</tr>
</thead>
</table>
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans Amazon Redshift.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Scénario associé

Pour un scénario associé, consultez Scénario : Charger/retirer des données dans/de Amazon S3 à la page 3303.
tRedshiftRollback

Ce composant permet d’éviter le commit de transaction involontaire.
Le tRedshiftRollback annule la transaction dans une base de données connectée.

Propriétés du tRedshiftRollback Standard

Ces propriétés sont utilisées pour configurer le tRedshiftRollback s’exécutant dans le framework de Jobs Standard.
Le composant tRedshiftRollback Standard appartient aux familles Cloud et Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

| Database | Sélectionnez un type de base de données dans la liste et cliquez sur Apply. |
| Component list | Sélectionnez le composant tRedshiftConnection dans la liste s’il y a plus d’une connexion dans votre Job. |
| Close Connection | Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Amazon Redshift, notamment les composants tRedshiftConnection et tRedshiftCommit. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes |
bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Scénario associé

Pour un scénario du même type, consultez **Scénario : Annuler l’insertion de données dans des tables mère/fille** à la page 2623.
tRedshiftRow

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler), selon la nature de la requête et de la base de données.

Le SQLBuilder vous aide à rapidement et aisément écrire vos requêtes.

Le tRedshiftRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

**Propriétés du tRedshiftRow Standard**

Ces propriétés sont utilisées pour configurer le tRedshiftRow s’exécutant dans le framework de Jobs Standard.

Le composant tRedshiftRow Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
2. **Basic settings** du composant de connexion créant cette connexion.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

### Host
Adresse IP du serveur de base de données.

### Port
Numéro du port d’écoute du serveur de base de données.

### Database
Nom de la base de données.

### Schema
Nom exact du schéma.

### Username et Password
Informations d’authentification de l’utilisateur de base de données.

Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

### Additional JDBC Parameters
Définissez des propriétés JDBC supplémentaires pour la connexion que vous créez. Les propriétés sont séparées par une espace et chaque propriété est une paire clé-valeur. Par exemple, `ssl=true & sslfactory=com.amazon.redshift.ssl.NonValidatingFactory`, qui signifie que la connexion sera créée en utilisant le SSL.

### Schema et Edit Schema
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

- **Repository** : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th><strong>Table Name</strong></th>
<th>Nom de la table de base à lire.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query type</strong></td>
<td>La requête peut être <strong>Built-in</strong> ou distante dans le <strong>Repository</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez la requête appropriée dans le Repository. Le champ <strong>Query</strong> est renseigné automatiquement.</td>
</tr>
<tr>
<td><strong>Guess Query</strong></td>
<td>Cliquez sur le bouton <strong>Guess Query</strong> pour générer la requête correspondant au schéma de votre table dans le champ <strong>Query</strong>.</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien <strong>Row &gt; Rejects</strong>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>Propagate QUERY’s recordset</strong></th>
<th>Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste <strong>use column</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remarque :</strong></td>
<td>Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type <strong>Object</strong>. Ce composant est généralement suivi du <strong>tParseRecordSet</strong>.</td>
</tr>
<tr>
<td><strong>Use PreparedStatement</strong></td>
<td>Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau <strong>Set PreparedStatement Parameter</strong>, définissez les valeurs des paramètres représentés par des ‘?’ dans l’instruction SQL définie dans le champ <strong>Query</strong> de l’onglet <strong>Basic settings</strong>.</td>
</tr>
</tbody>
</table>
### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exem...</td>
</tr>
</tbody>
</table>
ple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez : **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

---

**Limitation**

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

---

**Scénarios associés**

Pour des scénarios du même type, consultez :

- **Scénario : Combiner deux flux pour une sortie sélective** à la page 2706.
- **Procédure**.
- **Scénario : Supprimer et re-générer un index de table MySQL** à la page 2700.
Ce composant retire des données dans Amazon Redshift et les ajoute dans des fichiers dans Amazon S3.

Le composant tRedshiftUnload exécute une requête spécifiée dans Amazon Redshift et retire le résultat de la requête pour l’ajouter à un ou plusieurs fichiers dans Amazon S3.

Propriétés du tRedshiftUnload Standard

Ces propriétés sont utilisées pour configurer le tRedshiftUnload s’exécutant dans le framework de Jobs Standard.

Le composant tRedshiftUnload Standard appartient aux familles Cloud et Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In: Propriété utilisée ponctuellement.</td>
<td></td>
</tr>
<tr>
<td>Repository: Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Host</td>
<td>Saisissez l’adresse IP ou le nom de l’hôte du serveur de la base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute du serveur de la base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Saisissez le nom du schéma.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les données d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Additional JDBC Parameters</td>
<td>Définissez des propriétés JDBC supplémentaires pour la connexion que vous créez. Les propriétés sont séparées par une espace et chaque propriété est une paire clé-valeur. Par exemple, ssl=true &amp; ssifactor</td>
</tr>
</tbody>
</table>
Table Name

Saisissez le nom de la table de laquelle lire les données.

Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Query Type et Query

Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.

Guess Query

Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.

Access Key

Spécifiez l'ID de la clé d'accès identifiant de manière unique un compte AWS. Pour plus d'informations concernant l'obtention de votre clé d’accès et clé secrète, consultez Obtention de vos clés d’accès AWS.

Secret Key

Spécifiez la clé secrète d’accès, constituant les informations de sécurité, ainsi que la clé d'accès. Pour saisir la clé secrète, cliquez sur le bouton `[...]` à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres.
<table>
<thead>
<tr>
<th><strong>Bucket</strong></th>
<th>Saisissez le nom du bucket Amazon S3, le dossier de plus haut niveau, duquel retirer les données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key prefix</strong></td>
<td>Saisissez le préfixe du nom pour les fichiers à retirer de Amazon S3. Par défaut, les fichiers retirés sont écrits par slice du cluster Redshift et les noms de fichiers sont écrits au format <code>&lt;object_path&gt;/&lt;name_prefix&gt;&lt;slice-number&gt;_part_&lt;file-number&gt;</code>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| **File type** | Sélectionnez le type des fichiers retirés de Amazon S3 dans la liste :
| | • Delimited file or CSV : fichier délimité/CSV.
| | • Fixed width : fichier à taille fixe. |
| **Fields terminated by** | Saisissez le caractère utilisé pour séparer les champs.
| | Ce champ est disponible uniquement lorsque l'option Delimited file or CSV est sélectionnée dans la liste File type. |
| **Enclosed by** | Sélectionnez le caractère par lequel entourer les champs.
| | Cette liste est disponible uniquement lorsque l'option Delimited file or CSV est sélectionnée dans la liste File type. |
| **Fixed width mapping** | Saisissez une chaîne de caractères spécifiant un libellé et une largeur de colonne personnalisés entre guillemets doubles. Le format de la chaîne de caractères est :
| | ColumnLabel1:ColumnWidth1,ColumnLabel2:ColumnWidth2,...
| | Notez que le libellé de colonne dans la chaîne de caractères n'a aucune relation avec la nom de la colonne de la table et peut être une chaîne de caractères ou un entier. L'ordre des paires libellé/largeur doit correspondre à l'ordre des colonnes des tables.
| | Ce champ est disponible uniquement lorsque l'option Fixed width est sélectionnée dans la liste File type. |
| **Compressed by** | Cochez cette case et, dans la liste qui s'affiche, sélectionnez le type de compression des fichiers. |
| **Encrypt** | Cochez cette case pour crypter les fichiers retirés à l'aide du cryptage Amazon S3 côté client. Dans le champ Encryption key affiché, saisissez la clé de chiffrage utilisée pour chiffrer les fichiers retirés. Notez que seule une enveloppe de clé AES 128 bits ou AES 256 bits encodée en base64 est supportée. Pour plus d'informations, consultez Déchargement de fichiers de données chiffrés (en anglais). |
Specify null string
Cochez cette case et, dans la liste qui s’affiche, sélectionnez une chaine de caractères représentant une valeur nulle dans les fichiers retirés.

Escape
Cochez cette case pour placer un caractère d’échappement (\) avant chaque occurrence des caractères suivants pour les colonnes CHAR et VARCHAR dans les fichiers délimités retirés : interligne (\n), retour chariot (\r), le caractère séparateur spécifié pour les données retirées, le caractère d’échappement (\), un guillemet (" ou ’).

Overwrite s3 object if exist
Cochez cette case pour écraser les fichiers d’objets existant dans Amazon S3.

Parallel
Cochez cette case pour écrire les données en parallèle dans plusieurs fichiers retirés dans Amazon S3 selon le nombre de slices dans le cluster Redshift.

tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

Global Variables

Global Variables

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaine de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation
Ce composant couvre toutes les requêtes SQL possibles pour la base de données Amazon Redshift.

Dynamic settings
Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exem
ple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Pour un scénario associé, consultez Scénario : Charger/retirer des données dans/de Amazon S3 à la page 3303.
tReplace

Ce composant nettoie tous les fichiers avant traitement.

Le composant tReplace effectue un Rechercher/Remplacer dans les colonnes d’entrée spécifiées.

Propriétés du tReplace Standard

Ces propriétés sont utilisées pour configurer le tReplace s’exécutant dans le framework de Jobs Standard.

Le composant tReplace Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
Deux colonnes en lecture seule sont automatiquement ajoutées au schéma de sortie : Value et Match. |
| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |
| Simple Mode Search / Replace | Cliquez sur le bouton [+] pour ajouter autant de conditions que vous le souhaitez. Les conditions sont exécutées l’une après l’autre pour chaque ligne. |
**tReplace**

<table>
<thead>
<tr>
<th><strong>Input column</strong></th>
<th>Sélectionnez la colonne du schéma sur laquelle effectuer le Rechercher/Remplacer.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Search</strong></td>
<td>Saisissez la valeur à chercher dans la colonne d’entrée.</td>
</tr>
<tr>
<td><strong>Replace with</strong></td>
<td>Saisissez la valeur de remplacement.</td>
</tr>
<tr>
<td><strong>Whole word</strong></td>
<td>Cochez cette case si la valeur recherchée constitue un mot entier.</td>
</tr>
<tr>
<td><strong>Case sensitive</strong></td>
<td>Cochez cette case pour tenir compte de la casse.</td>
</tr>
<tr>
<td>Notez que vous ne pouvez pas utiliser d’expression régulière dans ces colonnes.</td>
<td></td>
</tr>
</tbody>
</table>

**Use advanced mode**

Cochez cette case si vous souhaitez utiliser des expressions régulières.

**Advanced settings**

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

<table>
<thead>
<tr>
<th><strong>Global Variables</strong></th>
<th><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>NB_LINE</strong> : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

**Utilisation**

| **Règle d’utilisation** | Ce composant n’est pas un composant de début puisqu’il requiert un flux de données. Il requiert aussi un composant de sortie. |
Scénario : Remplacement multiple et filtrage de colonnes

Le Job suivant recherche et remplace différentes coquilles et fautes dans un fichier .CSV puis filtre une colonne avant de générer en sortie un nouveau fichier .CSV contenant les données modifiées.

1. Cliquez et déposez les composants suivants dans l'espace de modélisation graphique : tFileInputDelimited, tReplace, tFilterColumn et tFileOutputDelimited.
2. Connectez les composants à l'aide de liens de type Main Row via un clic-droit sur chaque composant.
3. Sélectionnez le composant tFileInputDelimited et paramétrez les propriétés du flux d'entrée.

   - Property Type : Built-In
   - File Name : 'd:/Input/replace.csv'
   - Row Separator : "\n" Field Separator : "," Header : 0 Footer : 0 Limit : 0
   - Schema : Built-In, Edit schema, ... Skip empty rows
   - Encoding Type : ISO-8859-15

   - Le fichier est un simple fichier .csv stocké localement. Ainsi, dans le champ Filepath, renseignez le chemin d'accès au fichier. Dans le champ Row Separator, renseignez le séparateur de lignes, ici le retour chariot. Dans le champ Field Separator, renseignez le séparateur de champs, ici un point virgule. Dans cet exemple, l'en tête (Header) est le nom des colonnes, et il n'y a ni pied de page (Footer) ni de nombre limite de lignes à traiter (Limit).
4. Le fichier contient des caractères tels que : *t, $, . ou encore Nikson que l'on veut changer en Nixon, et streat, que l'on veut changer en Street.

<table>
<thead>
<tr>
<th>Street</th>
<th>FirstName</th>
<th>Name</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>streat</td>
<td>John</td>
<td>Kennedy</td>
<td>98.30$</td>
</tr>
<tr>
<td>streat</td>
<td>Richad</td>
<td>Nikson</td>
<td>78.23$</td>
</tr>
<tr>
<td>streat</td>
<td>Richard</td>
<td>Nikson</td>
<td>78.2$</td>
</tr>
<tr>
<td>streat</td>
<td>toto</td>
<td>Nikson</td>
<td>78.23$</td>
</tr>
<tr>
<td>streat</td>
<td>Richard</td>
<td>Nikson</td>
<td>78.23$</td>
</tr>
<tr>
<td>streat</td>
<td>Georges</td>
<td>bush</td>
<td>99.99$</td>
</tr>
</tbody>
</table>

   - Dans le champ Schema, sélectionnez aussi l'option Built-in. Le schéma est composé de quatre colonnes de différents types : chaînes de caractères (string) ou de nombres décimaux (float).
• Sélectionnez le composant **tReplace** pour paramétrer les propriétés des Rechercher/Remplacer.

<table>
<thead>
<tr>
<th>InputColumn</th>
<th>Search</th>
<th>Replace with</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Street</td>
<td>&quot;street&quot;</td>
<td>'Street'</td>
</tr>
<tr>
<td>Amount</td>
<td>&quot;$&quot;</td>
<td>&quot;£&quot;</td>
</tr>
<tr>
<td>Name</td>
<td>&quot;Nikson&quot;</td>
<td>&quot;Nixon&quot;</td>
</tr>
<tr>
<td>FirstName</td>
<td>&quot;t&quot;</td>
<td></td>
</tr>
</tbody>
</table>

• Le schéma peut être synchronisé à partir du flux d’entrée.

• Cochez la case **Simple mode** car les paramètres de recherche peuvent être facilement définis sans avoir recours aux expressions régulières.

• Cliquez sur le signe [+] pour ajouter des lignes aux tableaux des paramètres.

• En premier paramètre, sélectionnez **amount** dans la colonne **Input column**. Dans la colonne **Search**, cherchez tous les **points** et dans la colonne **Replace** indiquez la valeur de remplacement, ici la **virgule**. Les valeurs doivent être mises entre guillemets.

• En deuxième paramètre, sélectionnez **Street** dans la colonne **Input column**. Dans la colonne **Search**, cherchez la chaîne de caractères **street**. Dans la colonne **Replace**, renseignez la valeur de remplacement, ici **Street**.

• En troisième paramètre, sélectionnez à nouveau **Amount** dans la colonne **Input column**, saisissez **$** dans la colonne **Search** et **£** dans la colonne **Replace**.

• En quatrième paramètre, sélectionnez **Name** dans la colonne **Input column**. Dans le champ **Search**, saisissez **Nikson** dans la colonne **Search** et **Nixon** dans la colonne **Replace**.

• En cinquième paramètre, sélectionnez **FirstName** dans la colonne **Input column**. Dans le champ **Search**, saisissez la chaîne de caractère *t*, et ne mettez rien entre guillemets.

• L’option **Advanced mode** n’est pas utilisée pour ce scénario.

• Sélectionnez le composant suivant : **tFilterColumn**.
• Le composant **tFilterColumn** contient un éditeur de schéma permettant de créer un schéma de sortie basé sur le nom des colonnes du schéma d’entrée. Dans cet exemple, ajoutez une nouvelle colonne *empty_field* et changez l’ordre des colonnes pour obtenir le schéma suivant : *empty_field*, *Firstname*, *Name*, *Street*, *Amount*.

• Cliquez sur **OK** pour valider.

• Double-cliquez sur le **tFileOutputDelimited** afin d’afficher sa vue **Component**.

  - **Property Type**: **Built-In** [✓]
  - **File Name**: `D:\Input\CleanOutputFile.csv`
  - **Row Separator**: `;` [✓]
  - **Include Header**: [☐]
  - **Schema**: **Built-In** [✓]
  - **Encoding Type**: **ISO-8859-15** [✓]

• Dans le champ **Schema**, sélectionnez l’option **Built-in** et cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

• Enregistrez le Job et exécutez-le en appuyant sur **F6**.

<table>
<thead>
<tr>
<th>John</th>
<th>Kennedy</th>
<th>Street</th>
<th>98,30€</th>
</tr>
</thead>
<tbody>
<tr>
<td>Richad</td>
<td>Nixon</td>
<td>Street</td>
<td>78,23€</td>
</tr>
<tr>
<td>Richard</td>
<td>Nixon</td>
<td>Street</td>
<td>78,23€</td>
</tr>
<tr>
<td>toto</td>
<td>Nixon</td>
<td>Street</td>
<td>78,23€</td>
</tr>
<tr>
<td>Richard</td>
<td>Nixon</td>
<td>Street</td>
<td>78,23€</td>
</tr>
<tr>
<td>Georges</td>
<td>bush</td>
<td>street</td>
<td>99,99€</td>
</tr>
</tbody>
</table>

La première colonne est vide, les caractères parasites ont été supprimés des autres colonnes, et *Nikson* a été remplacé par *Nixon*. La colonne *Street* a été déplacée, les décimales sont dorénavant séparées par une virgule et le symbole monétaire a été changé.
**tReplaceList**

Ce composant nettoie tous les fichiers avant traitement.

Le tReplaceList effectue un rechercher/remplacer dans les colonnes d'entrée spécifiées en fonction des données de référence (lookup).

**Propriétés du tReplaceList Standard**

Ces propriétés sont utilisées pour configurer le tReplaceList s'exécutant dans le framework de Jobs Standard.

Le composant tReplaceList Standard appartient à la famille Data Quality.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.
Deux colonnes en lecture seule, Value et Match sont automatiquement ajoutées au schéma de sortie.

**Avertissement :**
Le type de données défini dans les schémas doit être cohérent, c'est-à-dire qu'un entier ne peut être remplacé que par un entier en utilisant un entier comme champ de référence. Une valeur d'un type ne peut être remplacée par une valeur d'un autre type. |
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Lookup search column</td>
</tr>
<tr>
<td>Lookup replacement column</td>
</tr>
<tr>
<td>Column options</td>
</tr>
</tbody>
</table>
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Global Variables


### Utilisation

| Règle d’utilisation | Ce composant est un composant intermédiaire, il nécessite un composant d’entrée et un composant de sortie. |

### Scénario : Remplacement à partir d’un fichier de référence

Le Job suivant recherche et remplace une liste d’états par le code à deux lettres correspondant. Les codes correspondants proviennent d’un fichier de référence placé dans le flux lookup du Job.
**Construire le Job**

**Procédure**

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique: deux tFileInputDelimited, un tReplaceList et un tLogRow.

2. Reliez les deux tFileInputDelimited au tReplaceList à l'aide de liens Row > Main. Notez que le lien entre le composant de référence (le second tFileInputDelimited) et le tReplaceList apparaît comme lien Lookup.

3. Reliez le composant tReplaceList au tLogRow à l'aide d'un lien Row > Main.

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le premier tFileInputDelimited pour ouvrir sa vue Basic settings et configurer les paramètres du flux d’entrée principal, notamment le chemin d’accès au fichier à lire, ainsi que le nombre de lignes d’en-tête à ignorer.

Dans cet exemple, le fichier principal d’entrée fournit une liste de personnes et d’états des États-Unis. Voici un extrait du fichier :

```
name;state
Andrew Kennedy;Mississippi
Benjamin Carter;Louisiana
Benjamin Monroe;West Virginia
Bill Harrison;Tennessee
Calvin Grant;Virginia
Chester Harrison;Rhode Island
Chester Hoover;Kansas
Chester Kennedy;Maryland
Chester Polk;Indiana
Dwight Nixon;Nevada
Dwight Roosevelt;Mississippi
Franklin Grant;Nebraska
```


Selon la structure du fichier d’entrée principal, le schéma d’entrée doit contenir deux colonnes: name et state.
Cliquez sur **OK** pour fermer la boîte de dialogue et propager les modifications au composant suivant.

3. Configurez de la même façon les propriétés du second **tFileInputDelimited**.

Dans cet exemple, le fichier de référencefournit une liste d’états ainsi que leur code à deux lettres. Le schéma de référence doit donc comprendre deux colonnes : *state* et *code*.

4. Double-cliquez sur le **tReplaceList** pour ouvrir sa vue **Basic settings** et paramétrer l’opération à effectuer.
5. Dans la liste **Lookup search column**, sélectionnez la colonne à rechercher. Dans ce scénario, recherchez la colonne *state*.

6. Dans la liste **Lookup replacement column**, sélectionnez la colonne contenant les valeurs de remplacement, *code* pour les codes à deux lettres des états, dans cet exemple.

7. Dans la table **Column options**, cochez la case **Replace** pour la colonne *states*, afin de remplacer le nom des états par le code correspondant.

8. Dans le composant **tLogRow**, sélectionnez l’option **Table** pour une meilleure lisibilité de la sortie.

**Exécuter le Job**

**Procédure**

Sauvegardez le Job et appuyez sur **F6** pour l’exécuter.

Le nom des états a été remplacé par le code à deux lettres correspondant.
tReplicate

Ce composant duplique le schéma entrant en deux flux identiques.
Ce composant effectue différentes opérations sur le même schéma.

Propriétés du tReplicate Standard

Ces propriétés sont utilisées pour configurer le tReplicate s’exécutant dans le framework de Jobs Standard.
Le composant tReplicate Standard appartient à la famille Orchestration.
Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
Cliquez sur Sync columns pour récupérer le schéma du composant précédent dans le Job. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-In</td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Repository</td>
<td>Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métdonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant n’est pas un composant de début (fond vert) puisqu’il requiert un composant d’entrée et un composant de sortie. |
| Connections | Liens de sortie (de ce composant à un autre) :
Row : Main
Trigger : Run if, OnComponentOk, OnComponentError.

Liens d’entrée (d’un autre composant à un celui-ci) :
Row : Main, Reject.

Pour plus d’informations concernant les liens, consultez le Guide utilisateur du Studio Talend. |

**Scénario : Répliquer un flux et trier deux flux identiques**

Ce scénario décrit un Job lisant un flux d’entrée contenant des noms et des états dans un fichier CSV, réplique le flux d’entrée, puis trie les deux flux identiques selon le nom et l’état respectivement et affiche les données triées dans la console.
Construire le Job

Procédure

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : un tFileInputDelimited, un tReplicate, deux tSortRow et deux tLogRow.
2. Reliez le tFileInputDelimited au tReplicate à l'aide d'un lien Row > Main.
3. Répétez les étapes ci-dessus afin de connecter le composant tReplicate aux deux tSortRow puis les deux tSortRow aux deux tLogRow (un chacun).
4. Renommez les composants afin de mieux identifier leur rôle au sein du Job.

Configurer les composants

Procédure

1. Double-cliquez sur le tFileInputDelimited pour ouvrir sa vue Basic settings.

2. Cliquez sur le bouton [...] à côté du champ File name/Stream afin de parcourir votre système jusqu'au fichier duquel vous souhaitez lire le flux d'entrée. Dans cet exemple, le fichier d'entrée se nomme Names&States.csv et contient deux colonnes : name et state.

```csv
name,state
Andrew Kennedy;Mississippi
Benjamin Carter;Louisiana
Benjamin Monroe;West Virginia
Bill Harrison;Tennessee
Calvin Grant;Virginia
```
3. Renseignez les champs **Header, Footer** et **Limit** selon vos besoins. Dans cet exemple, saisissez 1 dans le champ **Header** pour ignorer la première ligne du fichier d’entrée.

4. Cliquez sur **Edit schema** pour définir la structure des données du flux d’entrée.

![Edit schema](image)

5. Double-cliquez sur le premier **tSortRow** pour ouvrir sa vue **Basic settings**.

![Sort by name](image)

6. Dans la table **Criteria**, cliquez sur le bouton [+] pour ajouter une ligne et configurez les paramètres de tri pour la colonne du schéma à traiter. Pour trier les données d’entrée selon le nom, sélectionnez *name* pour **Schema column**. Sélectionnez *alpha* comme type de tri et *asc* comme ordre de tri.

   Pour plus d’informations concernant ces paramètres, consultez **tSortRow** à la page 3794.

7. Double-cliquez sur le second **tSortRow** et répétez l’étape ci-dessus pour configurer les paramètres pour la colonne contenant les états.
8. Dans la vue **Basic settings** de chaque **tLogRow**, sélectionnez **Table** dans la zone **Mode** pour un meilleur affichage des résultats d’exécution du Job.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Exécutez le Job en appuyant sur **F6** ou en cliquant sur le bouton **Run** de la vue **Run**.
Les données triées par nom et les données triées par état sont affichées dans la console.
tREST

Ce composant sert de client d’un service Web REST.
Le composant tREST envoie des requêtes HTTP à un fournisseur de services Web REpresentational State Transfer (REST) et obtient les réponses correspondantes.

Propriétés du tREST Standard

Ces propriétés sont utilisées pour configurer le tREST s’exécutant dans le framework de Jobs Standard.
Le composant tREST Standard appartient à la famille Internet.

⚠️ Avertissement :

Pour construire des Jobs nécessitant déploiement dans Talend Runtime, il est recommandé d’utiliser le composant tRESTClient, qui convient mieux à Talend Runtime.

Pour plus d’informations concernant le composant tRESTClient, consultez tRESTClient à la page 3379.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le composant tREST utilise toujours un schéma local (built-in) en lecture seule qui contient deux colonnes :
- Body : stocke les résultats provenant du serveur

⚠️ Avertissement :

Modifier le schéma peut engendrer une perte de la structure du schéma et donc un échec du composant.

| URL | Renseignez l’adresse URL du serveur Web REST à invoquer.

| HTTP Method | À partir de cette liste, sélectionnez une méthode HTTP qui décrit l’action souhaitée. Le sens spécifique des méthodes HTTP est soumis aux définitions de votre fournisseur de services Web. Ci-dessous, les définitions des méthodes HTTP généralement admises :
- GET : récupère les données du côté serveur en fonction des paramètres donnés. |
- **POST** : crée et télécharge les données en fonction des paramètres donnés.
- **PUT** : met à jour les données en fonction des paramètres donnés, ou crée les données si elles sont inexistantes.
- **DELETE** : supprime les données en fonction des paramètres donnés.

### HTTP Headers
Saisissez les paires nom/valeur *(name/value)* pour les en-têtes HTTP afin de définir les paramètres de l'opération HTTP demandée.


### HTTP Body
Saisissez les données utiles (payload) à charger du côté serveur lorsque les actions POST ou PUT sont sélectionnées.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant. |

### Global Variables

| **Global Variables** | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

### Utilisation

| **Règle d’utilisation** | Ce composant est utilisé comme client d’un service Web REST afin de communiquer avec un fournisseur de services Web REST. Il requiert un composant de sortie. |
| **Limitation** | L’exécution de JRE 1.6 est requise pour faire fonctionner ce composant. |
Scénario : Créer et récupérer des données en invoquant un service Web REST

Ce scénario décrit un Job simple qui invoque un service Web REST afin de créer un nouvel enregistrement client du côté serveur et de récupérer les informations du client. Une fois exécuté, ce Job affiche les informations correspondantes dans la console de la vue Run.

- A partir de la Palette, cliquez-déposez les composants suivants dans l’espace de modélisation graphique : deux composants tREST et deux composants tLogRow. Renommez les deux composants tREST afin qu’ils décrivent au mieux les actions à effectuer.
- Connectez chaque composant tREST à un tLogRow à l’aide d’une connexion de type Row > Main.
- Connectez le premier composant tREST au second tREST à l’aide d’une connexion de type Trigger > OnSubjobOk.

- Double-cliquez sur le premier composant tREST afin d’afficher l’onglet Basic settings de sa vue Component.

  **Basic settings**
  - Schema: Built-In
  - URL: "http://192.168.0.30:8080/customerservice/customers"
  - HTTP Method: POST
  - HTTP Headers: "Content-Type" => "application/xml"
  - HTTP Body: "<Customer><name>Steven</name></Customer>"

- Renseignez le champ URL avec l’adresse URL du serveur Web que vous êtes sur le point d’invoquer. Notez que l’adresse URL fournie dans ce scénario est destinée essentiellement à la démonstration, ce n’est pas une adresse valide.
- A partir de la liste HTTP Method, sélectionnez POST afin d’envoyer une requête HTTP permettant de créer un nouvel enregistrement.
• Cliquez sur le bouton [+] afin d’ajouter une ligne au tableau HTTP Headers, et saisissez la paire nom-valeur (name-value) adéquate, soumise aux définitions de votre fournisseur de service, en indiquant le type de média des données utiles (payload) à envoyer du côté serveur. Dans ce scénario, saisissez Content-Type et application/xml. Pour plus d’informations concernant les types de média Internet, consultez http://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.7 (en anglais).

• Saisissez dans le champ HTTP Body les données utiles (payload) à charger du côté serveur, à savoir <Customer><name>Steven</name></Customer> dans ce scénario, pour créer un enregistrement d’un nouvel utilisateur appelé Steven.

  Remarque :
  Si vous souhaitez inclure des guillemets double dans votre payload, assurez-vous d’utiliser un caractère d’échappement, la barre oblique inversée, avant chaque guillemet. Dans ce scénario par exemple, saisissez <Customer><name>"Steven"</name></Customer> si vous souhaitez mettre le prénom Steven entre guillemets double.

• Double-cliquez sur le second tREST afin d’afficher l’onglet Basic settings de sa vue Component.

• Renseignez le champ URL avec la même adresse URL que celle utilisée dans le premier composant tREST.

• A partir de la liste HTTP Method, sélectionnez GET afin d’envoyer une requête HTTP permittant de récupérer tous les enregistrements existants.

• Dans l’onglet Basic settings de la vue Component de chaque tLogRow, cochez la case Print component unique name in front of each output row (Afficher le nom unique du composant en face de chaque ligne de sortie) pour identifier au mieux les deux flux de sortie.

• Sauvegardez votre Job et appuyez sur F6 pour l’exécuter.

La console indique que le premier composant tREST envoie une requête HTTP côté serveur afin de créer un nouvel utilisateur appelé Steven et que le second composant tREST lit avec succès les données côté serveur comprenant les informations de l’utilisateur nouvellement créé.
Starting job Feature_17587_TREST at 11:34 22/02/2011.

[statistics] connecting to socket on port 3761
[statistics] connected
[LogRow_1] Body: <!--xml version='1.0' encoding='utf-8' standalone='yes'-->
<!--Customer><id>114</id><name>steven</name></Customer>|ERROR_CDX
[LogRow_2] Body:
<!--Customer><id>114</id><name>steven</name></Customer><Customer><id>123</id><name>John</name></Customer></Customers>|ERROR_CDX.
[statistics] disconnected
job feature_17587_TREST ended at 11:34 22/02/2011. [exit code=0]
tRESTClient

Ce composant interagit avec des fournisseurs de services Web RESTful, en envoyant des requêtes HTTP et HTTPS, à l’aide de CXF (JAX-RS) et en obtenant les réponses correspondantes.

Ce composant s’intègre parfaitement dans Talend Runtime pour un support de HTTPS et, ultérieurement, le support de fonctionnalités de qualité de service.

Propriétés du tRESTClient Standard

Ces propriétés sont utilisées pour configurer le tRESTClient s’exécutant dans le framework de Jobs Standard.

Le composant tRESTClient Standard appartient à la famille ESB.

Le composant de ce framework est toujours disponible.

Basic settings

| URL | Saisissez l’URL du serveur REST à invoquer. Lorsque la case Use Service Locator est cochée, ce champ ne s’affiche pas et l’URL du serveur REST est obtenue automatiquement du serveur du Service Locator. |
| Relative Path | Saisissez le chemin d’accès relatif au serveur REST à invoquer.  
Par exemple, vous souhaitez accéder à http://localhost:8888/services/Customers/list :  
Si la case Use Service Locator est décochée : vous pouvez saisir la première partie de l’adresse dans le champ URL et la seconde partie dans le champ Relative Path. Par exemple, saisissez http://localhost:8888 dans le champ URL et /services/Cust omers/list dans le champ Relative Path. Vous pouvez également saisir le chemin d’accès complet au serveur REST dans le champ URL et laisser vide le champ Relative Path.  
Si la case Use Service Locator est cochée : la partie dans le champ URL est donnée par le Service Locator. Dans ce cas, vous devez connaître la partie URL et spécifier la suite dans le champ Relative Path. Elle dépend du service interrogé. Par exemple, dans le tRESTRequest, vous pouvez paramétrer l’endpoint REST Endpoint à http://localhost:8888/services et cocher la case Use Service Locator. Si vous souhaitez utiliser ce service, du côté du tRESTClient, vous devez spécifier / customers/list dans le champ Relative Path. |
| HTTP Method | Dans cette liste, sélectionnez une méthode HTTP décrivant l’action souhaitée. Les significations des méthodes HTTP sont sujets aux définitions de votre fournisseur de service Web. Les définitions des méthodes HTTP généralement acceptées sont listées ci-dessous : |
- **GET** : récupère des données du côté serveur selon les paramètres donnés.
- **POST** : charge des données dans le serveur selon les paramètres donnés.
- **PUT** : met à jour des données selon les paramètres donnés, ou crée les données, si elles n’existent pas.
- **PATCH** : modifie partiellement les données selon les paramètres donnés.
- **DELETE** : supprime les données selon les paramètres donnés.

**Content Type**

Sélectionnez **XML, JSON ou FORM** selon le type de média du contenu à charger dans le serveur.

Cette liste apparaît lorsque vous sélectionnez la méthode HTTP **POST, PUT ou PATCH**.

**Accept Type**

Sélectionnez le type de média que le côté client est préparé à accepter pour la réponse, du côté serveur.

Les options disponibles sont : **XML, JSON et ANY**.

Lorsque **ANY** est sélectionnée, le message de réponse peut être de tout type et sera transformé en une chaîne de caractères (string).

**Query parameters**

Spécifiez les paramètres de la requête sous forme de paires nom-valeur.

Cette option est généralement utilisée avec la méthode GET.

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant.

Ce composant utilise trois schémas built-in, en lecture seule.

Cliquez sur **Edit Schema** pour visualiser la structure du schéma.

**Avertissement :**

Modifier le type du schéma peut conduire à une perte de cette structure et à un échec de l'exécution du composant.

**Input Schema**

Schéma des données d'entrée. Le schéma contient deux colonnes :

- **body** : stocke le contenu des données d'entrée structurées
- **string** : stocke le contenu d'entrée lorsqu'il est de type string, ou est géré comme tel.

**Response Schema**

Schéma pour la réponse du serveur. Ce schéma est passé au composant suivant à l'aide d’un lien **Row > Response** et contient trois colonnes :

- **statusCode** : stocke le code de statut HTTP du serveur.
<table>
<thead>
<tr>
<th><strong>tRESTClient</strong></th>
</tr>
</thead>
</table>
| - **body** : stocke le contenu d'une réponse structurée du serveur.  
- **string** : stocke le contenu de la réponse du serveur, lorsqu'elle est de type string ou gérée comme telle. |
| **Error Schema** |
| Schéma pour les informations d'erreur. Ce schéma est passé au composant suivant via un lien **Row > Error** et contient deux colonnes :  
- **errorCode** : stocke le code de statut HTTP du serveur lorsqu'une erreur survient durant le processus d'invocation. L'interprétation des codes d'erreur spécifiques est soumise aux définitions de votre fournisseur de services Web. Pour des informations de référence, consultez le site http://fr.wikipedia.org/wiki/Liste_des_codes_HTTP.  
- **errorMessage** : stocke le message d'erreur correspondant au code d'erreur. |
| **Use Service Locator** |
| Cochez cette case pour activer le Service Locator. Cela maintient la disponibilité du service et permet de répondre aux demandes et de respecter les (SLAs). Spécifiez l'espace de noms du Service ainsi que le nom du Service dans les champs correspondants. |
| **Use Service Activity Monitor** |
| Cochez cette case pour activer le Service Activity Monitor. Il capture les événements et stocke les informations afin de faciliter les analyses en profondeur de l'activité des services et de suivre les messages via une transaction métier. Cette option peut être utilisée, entre autres, pour analyser le temps de réponse du service, identifier les modèles de trafic ou effectuer une analyse de cause racine. |
| **Use Authentication** |
| Cochez cette case si l'authentification est requise du côté du serveur REST. Sélectionnez le type d'authentification entre les suivants :  
- **Basic HTTP** : renseignez les champs **Username** et **Password** avec vos identifiants.  
- **HTTP Digest** : renseignez les champs **Username** et **Password** avec vos identifiants.  
- **SAML Token (ESB runtime only)** : renseignez les champs **Username** et **Password** avec vos identifiants.  
- **OAuth2 Bearer** : renseignez le champ **Bearer Token** avec une chaîne contenant vos identifiants, encodée en base64.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Use Business Correlation** |
| Cochez cette case pour créer un ID de corrélation dans ce composant. Vous pouvez spécifier un ID de corrélation ID dans le champ **Correlation Value**. Dans ce cas, l'ID de corrélation |
est passé au service qu’il appelle, afin que les appels en chaîne des services soient groupés sous cet ID de corrélation. Si vous laissez ce champ vide, cette valeur est générée automatiquement lors de l’exécution.

Lorsque cette option est activée, le tRESTClient extrait également l’ID de corrélation de l’en-tête de réponse et le stocke dans la variable du composant pour usage ultérieur dans le flux.

<table>
<thead>
<tr>
<th>Die on error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Log messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour enregistrer l’échange de messages entre le service et le consommateur.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable WebClient Operation Reporting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si cette case est cochée, une propriété endpoint CXF est définie côté client. Cette propriété peut être utilisée par des intercepteurs métriques CXF de sortie afin de suivre les requêtes sur les services utilisant leurs URL. La propriété définie est org.apache.cxf.resource.operation.name et contient l’URL du service ainsi que le nom de la méthode HTTP, par exemple GET:<a href="http://example.com:8080/service/">http://example.com:8080/service/</a>. Si la case est décochée, la propriété n’est pas définie et certaines métriques ne sont pas comptées, ce qui a pour effet d’améliorer la performance.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Convert Response To DOM Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case afin de convertir la réponse du serveur en type Document. Décochez cette case si vous souhaitez que la réponse soit traitée comme une chaîne de caractères.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Convert Types To Strings</th>
</tr>
</thead>
</table>
| Cette option est disponible lorsque, dans la liste Content Type le type sélectionné est JSON, ou lorsque dans la liste Accept Type, le type sélectionné est JSON ou any. Cochez cette case pour convertir toutes les valeurs de JSON à String, entre guillemets doubles. Par exemple, une fois cette option activée, les données JSON :

```
"root": {
    "test": 111
}
```

 seront modifiées en :

```
"root": {
    "test": "111"
}
```
<table>
<thead>
<tr>
<th><strong>Drop JSON Request Root</strong></th>
<th>Cette option apparaît lorsque vous avez sélectionné POST, PUT ou PATCH dans la liste HTTP Method et JSON dans la liste Content Type. Cochez cette case pour supprimer les éléments JSON racine.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Wrap JSON Response</strong></td>
<td>Cette option est disponible et activée par défaut lorsque l’option JSON est sélectionnée dans la liste Accept Type de la vue Basic settings. Lorsque cette case est cochée, la réponse est entourée d’un élément root. Décocochez cette case si vous souhaitez supprimer cet élément root de la réponse.</td>
</tr>
<tr>
<td><strong>HTTP Headers</strong></td>
<td>Saisissez la (les) paire(s) nom-valeur des en-têtes HTTP pour définir les paramètres de l’opération HTTP demandée. Pour les définitions spécifiques des en-têtes HTTP, consultez votre fournisseur de services Web REST. Pour des informations de référence, consultez en.wikipedia.org/wiki/List_of_HTTP_headers (en anglais).</td>
</tr>
<tr>
<td><strong>Disable chunked encoding</strong></td>
<td>Cette option apparaît uniquement lorsque la méthode dans la liste HTTP Method est POST, PUT ou PATCH. Cochez cette case pour désactiver l’encodage des payloads par morceaux.</td>
</tr>
<tr>
<td><strong>Service Locator Customer Properties</strong></td>
<td>Cette table apparaît lorsque la case Use Service Locator est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ Property Name et Property Value respectivement.</td>
</tr>
<tr>
<td><strong>Service Activity Customer Properties</strong></td>
<td>Cette table apparaît lorsque la case Use Service Activity Monitor est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ Property Name et Property Value respectivement.</td>
</tr>
<tr>
<td><strong>Connection timeout</strong></td>
<td>Configurez le temps, en secondes, avant suspension, durant lequel le client attend pour établir une connexion. Si ce champ est configuré à 0, le client continue indéfiniment à essayer d’ouvrir une connexion. Cette option fonctionne uniquement dans le studio. Pour l’utiliser après le déploiement du composant dans le moteur d’exécution, dans le fichier de configuration &lt;TalendRuntimePath&gt;/container/etc/org.apache.cxf.http.conduits-common.cfg, ajoutez url = http://.* pour gérer les requêtes HTTP et HTTPS. Spécifiez ensuite le paramètre client.ConnectionTimeout de délai avant suspension en millisecondes. Si vous devez utiliser l’option Receive time out, spécifiez le délai client.ReceiveTimeout en millisecondes également.</td>
</tr>
<tr>
<td><strong>Receive timeout</strong></td>
<td>Configurez le temps, en secondes, durant lequel le client attend pour obtenir une réponse, avant suspension de la connexion. Si ce champ est configuré à 0, le client attend indéfiniment. Cette option fonctionne uniquement dans le studio. Pour l’utiliser après le déploiement du composant dans le moteur d’exécution ou dans <em>Talend Administration Center</em>, consultez l’option <em>Connection timeout</em>.</td>
</tr>
<tr>
<td><strong>Use HTTP proxy</strong></td>
<td>Cochez cette case si vous utilisez un serveur proxy. Une fois cochée, vous devez saisir les informations de connexion : hôte, port, identifiant et mot de passe.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

| **Global Variables** | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable *After* et retourne un entier.  
**HEADERS** : en-têtes des réponses HTTP. Cette variable est une variable *Flow* et retourne une liste des valeurs des en-têtes des réponses HTTP.  
**CORRELATION_ID** : l’ID de corrélation par lequel regr ouper les appels de services en chaîne. Cette variable est une variable *Flow* et retourne une chaîne de caractères.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  
Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches `Ctrl+Espace` pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

**Utilisation**

| **Règle d’utilisation** | Ce composant est utilisé comme client de service Web RESTful pour communiquer avec un fournisseur de services RESTful, avec la possibilité de saisir une requête dans un service, à l’intérieur d’un Job et de retourner le résultat du Job comme réponse du service. Selon les actions à effectuer, le composant peut être un composant de début ou intermédiaire. |
Cliquez sur le bouton [+] pour ajouter une ligne à la table et renseignez le champ Code avec une variable de contexte, afin d’activer ou désactiver dynamiquement l’option Use Authentication ou Use HTTP proxy lors de l’exécution. Vous pouvez ajouter deux lignes à la table afin de configurer les deux options.

Une fois qu’un paramètre dynamique est défini, l’option correspondante est mise en relief et devient inutilisable dans les vues Basic settings et Advanced settings.


Liens de sortie :
Row : Response ; Error.
Trigger : OnSubjobOk ; OnSubjobError ; Run if ; OnComponentOk ; OnComponentError.

Liens d’entrée :
Row : Main ; Reject.
Trigger : Run if ; OnSubjobOk ; OnSubjobError ; OnComponentOk ; OnComponentError.

Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.

L’utilisation des variables de contexte pour les endpoints dynamiques ou les espaces de noms du Service Locator fonctionne uniquement dans le Studio. Elle n’est pas supportée dans le Runtime.

Scénario 1 : Obtenir des informations sur un utilisateur en interagissant avec un service RESTful

Ce scénario décrit un Job composé de trois composants qui récupère des informations sur un utilisateur en se basant sur l’identifiant de l’utilisateur depuis un service REST via HTTP GET. Les informations récupérées, ainsi que l’échange de messages entre le client et le serveur, sont affichés dans la console Run.

Prérequis
Si vous êtes un utilisateur de la solution Talend Open Studio for ESB, créez un Job comme décrit dans Scénario 2 : Utiliser les paramètres URI Query pour explorer les données d’une base de données à la page 3409, exécutez le Job afin d’exposer un service REST et saisissez l’URL du service REST dans votre navigateur Web, http://localhost:8088/users dans cet exemple. Vous devriez voir des informations telles que les suivantes :

Row : Response ; Error.
Trigger : OnSubjobOk ; OnSubjobError ; Run if ; OnComponentOk ; OnComponentError.

Row : Main ; Reject.
Trigger : Run if ; OnSubjobOk ; OnSubjobError ; OnComponentOk ; OnComponentError.

Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.
Si vous n’êtes pas un utilisateur de la solution Talend Open Studio for ESB, vous devez obtenir l’URL, la structure des données et les paramètres requis par le service REST appelé depuis votre fournisseur de service REST. Vous devez également apporter les modifications nécessaires aux configurations présentées dans le scénario.

Construire le Job

Procédure

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique :
   - un tRESTClient, utilisé pour appeler le service REST et récupérer les informations sur l’utilisateur depuis le serveur,
   - un tXMLMap, utilisé pour adapter l’arborescence du service REST
   - et un tLogRow afin d’afficher les informations sur l’utilisateur récupérées dans la console Run.
2. Reliez le tRESTClient au tXMLMap à l’aide d’un lien Row > Response.
3. Reliez le premier tXMLMap au tLogRow à l’aide d’un lien Row > Main. Nommez cette connexion out dans cet exemple.
4. Reliez le second tXMLMap au tFileOutputDelimited à l’aide d’un lien Row > Main et renommez ce lien out.
5. Afin de mieux identifier le rôle de chaque composant, renommez-les comme suit :
Configurer les composants

Configurer l'appel du service

Procédure

1. Double-cliquez sur le **tRESTClient** pour ouvrir sa vue **Basic settings**.


3. Dans la liste **HTTP Method**, sélectionnez **GET** pour envoyer une requête HTTP afin de récupérer les enregistrements existants. Dans la liste **Accept Type**, sélectionnez le type accepté par le côté client pour la réponse du côté serveur, **XML**. Laissez les autres paramètres tels qu'ils sont.

4. Cliquez sur le bouton **[+]** situé sous la table **Query parameters** afin d'ajouter deux paramètres, **from** et **to**. Définissez la valeur de ces paramètres à 2 afin d'obtenir les informations de l'utilisateur ayant pour identifiant 2. Sinon, vous pouvez également obtenir les informations de l'utilisateur ayant pour identifiant 2 en ajoutant **?from=2&to=2** à l'URL du service.

5. Dans la vue **Advanced settings** du **tRESTClient**, cochez les cases **Log messages** et **Convert Response To DOM Document** afin d'enregistrer le contenu de l'échange de messages avec le serveur et convertir les réponses du serveur en type Document.
Mapper la structure du service et afficher les informations de l’utilisateur récupérées

**Procédure**

1. Double-cliquez sur le tXMLMap pour ouvrir l’éditeur de mapping.

2. Si vous avez sélectionné XML dans la liste Accept Type du composant tRESTClient, définissez la structure de l’arbre XML selon la structure du service.
   a) Dans la table d’entrée, dans la colonne body, cliquez-droit sur le nœud root. Dans le menu contextuel, cliquez sur Rename et renommez le nœud users.
   b) Cliquez-droit sur le nœud users. Dans le menu contextuel, cliquez sur Create Sub-Element et créez un sous-élément nommé users. Définissez l’élément user en tant qu’élément de boucle car la structure XML du service Web à invoquer effectue une boucle sur cet élément.
   c) Cliquez-droit sur le nœud user. Dans le menu contextuel, cliquez sur Create Attribute et saisissez id dans la boîte de dialogue [Create New Attribute] afin de créer un attribut nommé id pour le nœud user.
   d) Cliquez-droit à nouveau sur le nœud user. Dans le menu contextuel, cliquez sur Create Sub-Element et saisissez first_name dans la boîte de dialogue [Create New Element] afin de créer un sous-élément nommé first_name pour le nœud user.

Répétez cette opération afin de créer un autre sous-élément last_name sous le nœud user.
e) Déposez les colonnes `id`, `first_name` et `last_name` de la table d’entrée dans la table de sortie. Cliquez ensuite sur OK pour valider le mapping et fermer l’éditeur.

Si vous avez sélectionné **JSON** dans la liste **Accept Type** du composant **tRESTClient**, la réponse du serveur est renvoyée au format JSON et convertie en type Document. Dans cet exemple, la structure de la réponse convertie se présente comme suit :

```xml
<root>
  <users>
    <user>
      <id>2</id>
      <first_name>Theodore</first_name>
      <last_name>Harding</last_name>
    </user>
  </users>
</root>
```
Notez que l’élément `<root>` est supprimé si la case **Wrap JSON Response** est cochée dans la vue **Advanced settings** du composant **tRESTClient**.

Définissez la structure XML en conséquence et mappez-la comme décrit précédemment.

3. Double-cliquez sur le **tLogRow** afin d’ouvrir sa vue **Basic settings**.

4. Cliquez sur le bouton **Sync columns** afin de vous assurer que le schéma du composant est synchronisé avec le schéma de sortie du **tXMLMap**.

5. Dans le champ **Mode**, sélectionnez l’option **Table** afin d’afficher le résultat de l’opération GET sous forme de tableau.
Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur les touches Ctrl+S pour sauvegarder votre Job.
2. Appuyez sur la touche F6 ou sur le bouton Run de la vue Run pour exécuter le Job.

La console montre que le tRESTClient lit les informations sur l’utilisateur correspondant à l’identifiant spécifié depuis le serveur.

Si vous avez sélectionné XML dans la liste Accept Type du composant tRESTClient, les résultats d’exécution se présentent comme suit :

```
[statistics] connecting to socket on port 3496
[statistics] connected
May 26, 2014 5:38:17 PM
org.apache.cxf.interceptor.LoggingOutInterceptor
INFO: Outbound Message
------------------------
ID: 1
Address: http://localhost:8088/users?from=2&to=2
Http-Method: GET
Content-Type: */*
Headers: {Accept=[application/xml], Content-Type=[*/*]}
------------------------
May 26, 2014 5:38:17 PM
org.apache.cxf.interceptor.LoggingInInterceptor
INFO: Inbound Message
------------------------
ID: 1
Response-Code: 200
Encoding: ISO-8859-1
Content-Type: application/xml
Headers: {content-type=[application/xml], Date=[Mon, 26 May 2014 09:38:17 GMT], Server= [Jetty(6.1.14.v20131031)], transfer-encoding=[chunked]}
Payload: <?xml version="1.0" encoding="UTF-8"?><users><user id="2"><first_name>Theodore</first_name><last_name>Harding</last_name></user></users>
```

Si vous avez sélectionné JSON dans la liste Accept Type du composant tRESTClient, les résultats d’exécution se présentent comme suit :

```
[statistics] disconnected
```
Scénario 2 : Mettre à jour les informations des utilisateurs via une interaction avec un service RESTful

Ce scénario décrit un Job à trois composants mettant à jour, dans une base de données distante, les informations d’une liste d’utilisateurs via un service REST, à l’aide de la méthode HTTP POST. Une fois exécuté, le Job affiche les informations de l’échange serveur-client dans la console de la vue Run.

Les informations à mettre à jour sur le serveur sont stockées dans un fichier CSV, qui se présente comme suit :

<table>
<thead>
<tr>
<th>id;first_name;last_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1;John;Smith</td>
</tr>
<tr>
<td>2;Martin;Reagan</td>
</tr>
</tbody>
</table>
Prérequis:


```
<application>
  <grammars/>
  <resources base="http://localhost:8045/users">
    <resource path="/">
      <resource path="post/{id}/{first_name}/{last_name}">
        <param name="last_name" style="template" type="xs:string"/>
        <param name="id" style="template" type="xs:int"/>
        <param name="first_name" style="template" type="xs:string"/>
      </resource>
      <method name="POST">
        <request/>
        <response>
          <representation media-Type="application/xml"/>
          <representation media-Type="text/xml"/>
          <representation media-Type="application/json"/>
        </response>
      </method>
    </resource>
  </resources>
</application>
```

Si vous n’utilisez pas la solution *Talend Open Studio for ESB*, vous devez obtenir les informations relatives au service à partir de votre fournisseur de services REST, comme l’URL, le chemin d’accès à la ressource, ainsi que la structure des données. Vous devez également apporter au scénario les modifications nécessaires à vos configurations.
**Construire le Job**

**Procédure**

1. Créez un Job et ajoutez les composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette :
   - un **tFileInputDelimited**, pour lire les informations des utilisateurs à partir d’un fichier local,
   - un **tXMLMap**, pour adapter la structure d’entrée à la structure du service REST,
   - et un **tRESTClient**, utilisé pour appeler le service REST afin d’envoyer des données à la base de données distante.

2. Connectez le **tFileInputDelimited** au **tXMLMap** à l’aide d’un lien **Row > Main**.

3. Connectez le **tXMLMap** au **tRESTClient** à l’aide d’un lien **Row > Main** et renommez le flux de sortie, **request** dans cet exemple.

4. Renommez les composants afin mieux identifier leur rôle.

![Diagramme de connexion des composants](image)

**Configurer les composants**

**Configurer les données d’entrée et les mappings de structure**

**Procédure**

1. Double-cliquez sur le **tFileInputDelimited** pour ouvrir sa vue **Basic settings**.

   ![Interface de configuration du tFileInputDelimited](image)

   2. Spécifiez le fichier d’entrée dans le champ **File name**, renseignez le champ **Header** en saisissant 1 afin d’ignorer l’en-tête et laissez les autres paramètres tels qu’ils sont.

   3. Cliquez sur le bouton [...] à côté du champ **Edit schema** afin d’ouvrir la boîte de dialogue [Schema] et de modifier le schéma d’entrée comme suit. Ajoutez les colonnes suivantes :
      - **id**, de type **Integer**, de longueur (Length) 2. Cochez la case **Key** pour en faire la colonne clé,
      - **first_name**, de type **String**,
- `last_name`, de type `String`.

4. Double-cliquez sur le composant `tXMLMap` pour ouvrir son éditeur.

5. Renommez le nœud `root` dans la table de sortie : cliquez-droit sur le nœud, sélectionnez Rename dans le menu contextuel et spécifiez un nouveau nom dans la boîte de dialogue, `user` dans cet exemple.

6. Sélectionnez les trois colonnes de la table d’entrée et déposez-les sur le nœud `user`, puis sélectionnez l’option Create as sub-element of target node dans la boîte de dialogue afin de configurer ces colonnes comme des sous-éléments du nœud `user`. Cela fait, cliquez sur OK afin de valider les mappings et fermer l’éditeur.
Configurer l’appel du service

Procédure
1. Double-cliquez sur le composant tRESTClient pour ouvrir sa vue Basic settings.
2. Renseignez le champ **URL** avec l'emplacement de l'URI où est accessible le service REST, "http://localhost:8088/users" dans cet exemple.

3. Dans le champ **Relative Path**, saisissez le chemin d'accès à la ressource, "/post/" + row1.id + "/" + row1.first_name + "/" + row1.last_name dans cet exemple. Cela va envoyer les données de la ligne d'entrée au serveur via le chemin de la ressource.

4. Dans la liste **HTTP Method**, sélectionnez **GET** afin d'envoyer une requête HTTP pour récupérer les enregistrements existants.
   Dans la liste **Accept Type**, sélectionnez le type que le côté client accepte pour la réponse du côté serveur, **XML**.

5. Dans la vue **Advanced settings** du composant **tRESTClient**, cochez la case **Log messages** afin d'enregistrer les informations relatives à l'échange de messages avec le serveur.
   Laissez les autres paramètres tels qu'ils sont.

**Exécuter le Job et vérifier les résultats**

Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job et appuyez sur **F6** pour l'exécuter.

La console affiche les informations relatives à l'échange client serveur :
La console du Job utilisé en tant que serveur affiche les informations relatives à l'échange, ainsi que le résultat de la mise à jour de la base de données.
ID: 5
Address: http://localhost:8045/users/post/5/RoberThomson
Encoding: ISO-8859-1
Http-Method: POST
Content-Type: application/xml
Headers: {Accept=[/*/*], Cache-Control=[no-cache], connection=[keep-alive], Content-Length=[123], content-type=[application/xml], Host=[localhost:8045], Pragma=[no-cache], User-Agent=[Apache CXF 2.7.11]}
Payload: <user><id>5</id><first_name>Rober</first_name><last_name>Thomson</last_name></user>

<table>
<thead>
<tr>
<th>ID</th>
<th>first_name</th>
<th>last_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>Smith</td>
</tr>
<tr>
<td>2</td>
<td>Martin</td>
<td>Reagan</td>
</tr>
<tr>
<td>3</td>
<td>James</td>
<td>White</td>
</tr>
<tr>
<td>4</td>
<td>Jenny</td>
<td>Jackson</td>
</tr>
<tr>
<td>5</td>
<td>Robert</td>
<td>Thomson</td>
</tr>
</tbody>
</table>

[INFO ]: org.apache.cxf.interceptor.LoggingOutInterceptor - Outbound Message
ID: 5
Response-Code: 200
Content-Type: application/xml
Headers: {Content-Type=[application/xml], Date=[Thu, 29 May 2014 09:13:02 GMT]}
Payload: <user><id>5</id><first_name>Rober</first_name><last_name>Thomson</last_name></user>
tRESTRequest

Ce composant reçoit les requêtes GET/POST/PUT/PATCH/DELETE des clients, du côté serveur.

Le tRestRequest accepte les requêtes HTTP et/ou HTTPS des clients et supporte les méthodes HTTP GET, POST, PUT, PATCH et DELETE.

Remarque :
Pour activer le support de HTTPS, vous devez générer un keystore et ajouter quelques propriétés liées à la configuration de sécurité HTTPS dans le fichier `org.ops4j.pax.web.cfg` de votre conteneur Talend Runtime. Pour plus d’informations, consultez le Talend ESB Container Administration Guide (en anglais).

Propriétés du tRESTRequest Standard

Ces propriétés sont utilisées pour configurer le tRESTRequest s’exécutant dans le framework de Jobs Standard.

Le composant tRESTRequest Standard appartient à la famille ESB.

Ce composant est adapté pour une utilisation au sein de la perspective Mediation du Studio Talend. Il requiert l’utilisation du nœud du Repository Service et des assistants de création de Services.

Basic settings

| REST Endpoint | Renseignez ce champ en saisissant l’emplacement de l’URI où le service Web RESTful sera accessible pour les requêtes.

Remarque :
Si vous voulez que votre service soit disponible à la fois sur HTTP et HTTPS, renseignez ce champ avec un chemin relatif.

| REST API Mapping | Cliquez sur le bouton [+] sous la table de mapping pour ajouter des lignes pour spécifier des requêtes HTTP :

Output Flow : Cliquez sur le bouton [...] afin de spécifier le nom d’un flux de sortie et configurez le schéma du flux de sortie dans la boîte de dialogue qui suit.

Le schéma n’est pas obligatoire, donc, si vous ne devez pas passer des paramètres supplémentaires au composant tRESTRequest, vous pouvez laisser le schéma vide. Cependant, vous devez alimenter le schéma si vous avez des paramètres URI Path définis dans le champ URI Pattern ou si vous devez ajouter des paramètres facultatifs de requêtes, tels que URI Query, HTTP Header ou Form parameters à l’URI spécifiée dans le champ REST Endpoint.

Ajouter un schéma dont le nom est body permet de récupérer le corps de la requête des méthodes POST ou PUT. Les types Document, String et Byte[] sont supportés. |
Si vous spécifiez des paramètres d’URI dans le schéma du flux de sortie, vous devez définir le type de paramètre dans le champ Comment du schéma. Par défaut, si vous laissez le champ Comment vide, le paramètre est considéré comme un paramètre Path. Voici une liste des valeurs supportées dans Comment :

- vide ou path correspond au paramètre par défaut @PathParam,
- query correspond à @QueryParam,
- form correspond à @FormParam,
- header correspond à @HeaderParam.
- matrix correspond à @MatrixParam.
- multipart correspond @Multipart de CXF, représentant le corps de la requête. Il peut être utilisé avec la méthode POST ou PUT HTTP.

Remarque :
Il est recommandé de définir les valeurs par défaut de vos paramètres facultatifs (Header, Query, Form), pour ce faire, renseignez les colonnes Default du schéma.

HTTP Verb : Sélectionnez une méthode HTTP (GET/POST/PUT/PATCH/DELETE/OPTIONS/HEAD) dans la liste.

URI pattern : Renseignez ce champ avec les URIs RESTful décrivant la ressource.

Consumes : Sélectionnez dans la liste le type de format du contenu consommé que le composant utilise, entre XML or JSON, XML, JSON, Form, Multipart et Any lorsque la méthode HTTP est POST, PUT ou PATCH.

Produces : Lorsque la méthode HTTP est GET, POST, PUT, PATCH ou DELETE, sélectionnez dans la liste le type de format du contenu produit que le composant utilise, entre XML or JSON, XML, JSON, HTML et Any, ou sélectionnez <oneway> dans la liste pour accepter les requêtes à sens unique.

Streaming : Cochez cette case pour mettre en flux les données de réponse par morceaux, afin que les grands volumes de données puissent être traités efficacement.

Use Service Locator
Cochez cette case pour activer le Service Locator. Cela maintient la disponibilité du service et permet de répondre aux demandes et de respecter les (SLAs). Spécifiez l’espace de noms du Service ainsi que le nom du Service dans les champs correspondants.

Use Service Activity Monitor
Cochez cette case pour activer le Service Activity Monitor. Il capture les événements et stocke les informations afin de faciliter les analyses en profondeur de l’activité des services et de suivre les messages via une transaction métier. Cette option peut être utilisée, entre autres, pour analyser le temps de réponse du
service, identifier les modèles de trafic ou effectuer une analyse de cause racine.

| Use Authentication (ESB runtime only) | Cochez cette case pour activer l’option d’authentification dans ce service. Cette option fonctionne uniquement dans le Runtime. Sélectionnez le type d’authentification entre les suivants :

* Basic HTTP : la méthode la plus simple pour implémenter les contrôles d’accès aux ressources Web à l’aide de champs standard dans l’en-tête HTTP.
* SAML Token : format de données XML basé sur des standards ouverts pour échanger des données d’authentification et d’autorisation entre un fournisseur d’identité et un fournisseur de service. |

| Use Business Correlation | Cochez cette case pour activer l’option de corrélation, afin que les appels en chaîne de services soient groupés sous le même ID de corrélation ID. Le tRESTRequest extrait l’ID de corrélation de l’en-tête de la requête et le stocke dans la variable du composant pour usage ultérieur dans le flux.

Si cette option n’est pas activée du côté client, un ID de corrélation est automatiquement généré dans le tRESTRequest. |

**Advanced settings**

| Log messages | Cochez cette case pour enregistrer l’échange de messages entre le service et le consommateur. |

| Wrap JSON Request | Cochez cette case pour entourer la requête JSON d’un élément *root*. |

| Convert JSON values to String in response | Cochez cette case pour convertir les valeurs JSON en format String dans la réponse. |

| Service Locator Customer Properties | Cette table s’affiche lorsque la case Use Service Locator est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ Property Name et Property Value respectivement. |

| Service Activity Customer Properties | Cette table s’affiche lorsque la case Use Service Activity Monitor est cochée. Vous pouvez ajouter autant de lignes que nécessaire afin de personnaliser les propriétés correspondantes. Saisissez entre guillemets le nom et la valeur de chaque propriété dans le champ Property Name et Property Value respectivement. |

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |
### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>nombre de lignes traitées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>URI</strong></td>
<td>URI de la requête REST. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>URI_BASE</strong></td>
<td>URI de base de la requête REST. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>URI_ABSOLUTE</strong></td>
<td>URI absolue de la requête REST. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>HTTP_METHOD</strong></td>
<td>méthode HTTP. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ATTACHMENT_HEADERS</strong></td>
<td>en-têtes des pièces jointes de la requête REST. Cette variable est une variable <em>Flow</em> et retourne une liste des valeurs des en-têtes des pièces jointes.</td>
</tr>
<tr>
<td><strong>ATTACHMENT_FILENAMES</strong></td>
<td>nom des fichiers des pièces jointes de la requête REST. Cette variable est une variable <em>Flow</em> et retourne tous les noms de fichiers des pièces jointes.</td>
</tr>
<tr>
<td><strong>PRINCIPAL_NAME</strong></td>
<td>nom du Principal de la requête REST. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>CORRELATION_ID</strong></td>
<td>l'ID de corrélation par lequel recopier les appels de services en chaîne. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Règle d’utilisation</strong></td>
<td>Ce composant permet d’exposer un Job Talend en tant que service et de transmettre une requête à un service dans un Job puis retourner les résultats du Job en tant que réponse du service.</td>
</tr>
<tr>
<td></td>
<td>Le composant <strong>tRESTRequest</strong> doit être utilisé avec le <strong>tRESTResponse</strong> afin de fournir les résultats d’un Job en</td>
</tr>
</tbody>
</table>
tRESTRequest

| Limitation | Utilisation des variables de contexte pour les endpoints dynamiques ou les espaces de noms du Service Locator fonctionne uniquement dans le Studio. Elle n’est pas supportée dans le Runtime. |

Scénario 1 : Service REST acceptant des requêtes HTTP GET et envoyant des réponses

Ce scénario s’applique uniquement aux solutions Talend avec ESB.

Ce scénario décrit le processus d’acceptation d’une requête HTTP du client, son traitement et l’envoi de sa réponse.

Configurer le composant tRESTRequest

Procédure

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un tRESTRequest, deux tXMLMap et deux tRESTResponse.
2. Double-cliquez sur le composant tRESTRequest dans l’espace de modélisation graphique afin d’afficher sa vue Basic settings.

**Remarque :**
Si vous voulez que votre service soit disponible à la fois sur HTTP et HTTPS, renseignez ce champ avec un chemin relatif. Par exemple, si vous saisissez “/test”, votre service sera disponible à la fois sur http://<EndpointHTTPParDéfaut>/test et https://<EndpointHTTPSParDéfaut>/test, à condition que vous ayez configuré votre conteneur d’exécution pour supporter l’HTTPS. Pour plus d’informations, consultez le *Talend ESB Container Administration Guide* (en anglais).

4. Cliquez sur le bouton `[+]` pour ajouter une ligne dans la table **REST API Mapping**.

5. Sélectionnez la nouvelle ligne et cliquez sur le bouton `[...]` de la colonne **Output Flow** afin d’ajouter un schéma pour le flux de sortie.

Dans ce scénario, nommez le flux de sortie *GetOneUser*.


7. Sélectionnez GET dans la colonne **HTTP Verb**.

8. Dans la colonne **URI Pattern**, saisissez “/{id}/”.

9. De la même manière, ajoutez une ligne dans la table **REST API Mapping** et nommez ce schéma *GetUserNumber*. Ajoutez une ligne *string* de type *String* au schéma. Saisissez
query dans le champ **Comment** pour ajouter à la requête le paramètre des numéros.

Sélectionnez **GET** dans la liste de la colonne **HTTP Verb**. Renseignez le champ dans la colonne **URI Pattern** en saisissant "/number".

**Configurer le premier composant tXMLMap**

**Procédure**

1. Reliez le composant **tRESTRequest** au **tXMLMap** à l'aide d'un lien **Row > GetOneUser**.
2. Double-cliquez sur le **tXMLMap** dans l'espace de modélisation graphique pour ouvrir le **Map Editor**.

3. Cliquez sur le bouton [+ ] en haut à droite afin d’ajouter une table de sortie et nommez-la **ResponseUsers**.
4. Cliquez sur le bouton [+ ] en bas à droite pour ajouter deux colonnes en sortie.
   Nommez la première colonne **body** et, dans la colonne Type, sélectionnez **Document**.
   Nommez la seconde colonne **string** et, dans la colonne Type, sélectionnez **String**.
5. Cliquez-droit sur le nœud **root** et sélectionnez **Create Sub-Element** pour créer un sous-élément.
   Nommez le sous-élément **foo** dans la boîte de dialogue qui s’ouvre.
6. Cliquez-droit sur le nœud `foo` et sélectionnez **As loop element**.

7. Sélectionnez la colonne `id` de la table `GetOneUser` et déposez-la dans le champ **Expression** du nœud `foo` dans la table `ResponseUsers`.

![Diagramme de configuration de l'élément RESTRequest](image)

8. Cliquez sur **OK** afin de sauvegarder les paramètres.

**Configurer le second composant tXMLMap**

**Procédure**

1. Reliez le **tRESTRequest** à l’autre **tXMLMap** à l’aide d’un lien **Row > GetUserNumber**.

2. Double-cliquez sur le **tXMLMap** dans l’espace de modélisation graphique pour ouvrir le **Map Editor**.

![Diagramme de configuration de l'élément XMLMap](image)

3. Cliquez sur le bouton `[+]` en haut à droite afin d’ajouter une table de sortie et nommez-la `ResponseUserNumber`.

4. Cliquez sur le bouton `[+]` en bas à droite pour ajouter une colonne de sortie. Nommez la première colonne `body` et, dans la colonne **Type**, sélectionnez **Document**.

5. Cliquez-droit sur le nœud `root` et sélectionnez **Create Sub-Element** pour créer un sous-élément. Nommez le sous-élément `number` dans la boîte de dialogue qui s’ouvre.

6. Cliquez-droit sur le nœud `number` et sélectionnez **As loop element**.
7. Sélectionnez la colonne *string* de la table *GetUserNumber* et déposez-la dans le champ *Expression* du nœud *number* dans la table *ResponseUserNumber*.

8. Cliquez sur *OK* afin de sauvegarder les paramètres.

**Configurer le composant tRESTResponse**

**Procédure**

1. Reliez le *tXMLMap* à l’un des *tRESTResponse* à l’aide d’un lien *Row > ResponseUsers*.
   Le schéma défini dans le *tXMLMap* est automatiquement récupéré par le *tRESTResponse*.

   ![tRESTResponse_1](image)

2. Sélectionnez *OK(200)* dans la liste *Return status code*.
3. Laissez les autres paramètres tels qu’ils sont.
4. Reliez le *tRESTRequest* à l’autre *tRESTResponse* à l’aide d’un lien *Row > GetUserNumber*.

**Sauvegarder et exécuter le Job**

**Procédure**


```plaintext
[statistics] connecting to socket on port 3088
[statistics] connected
[INFO ]: org.apache.cxf.endpoint.ServerImpl - Setting the server's publish address to be http://localhost:8088/user
[INFO ]: org.eclipse.jetty.server.AbstractConnector - Started
Select:ChannelConnector@localhost:8088
```
2. Ouvrez votre navigateur si vous souhaitez tester le service.

La requête HTTP demandant l'ID d'un utilisateur est acceptée par le service REST et la réponse HTTP est retournée au serveur.


Vous pouvez constater que `123` est retourné dans la réponse.

**Scénario 2 : Utiliser les paramètres URI Query pour explorer les données d'une base de données**

Ce scénario s'applique uniquement aux solutions Talend avec ESB.

Ce scénario décrit comment utiliser les paramètres URI query dans le `tRESTRequest` afin d'explorer les données d'une base de données et d'envoyer la réponse via le `tRESTResponse`.

Pour ce faire, vous devez créer deux sous-jobs connectés à l'aide d'un lien `OnSubjobOk`. Ainsi, les deux sous-jobs seront exécutés séquentiellement. Pour plus d'informations concernant les connexions Trigger, consultez le **Guide utilisateur du Studio Talend**. Le premier sous-job crée et alimente la base de données. Le second permet d'explorer la base de données via le service REST.
Créer le premier sous-job

Pourquoi et quand exécuter cette tâche
Pour ce faire, procédez comme suit :

Procédure
1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : tFixedFlowInput de la famille Misc et tMysqlOutput de la famille Databases > Mysql.
2. Reliez le tFixedFlowInput au tMysqlOutput à l’aide d’un lien Row > Main.
3. Double-cliquez sur le tFixedFlowInput pour afficher sa vue Basic settings :

![Schema editeur.png](attachment:image.png)


6. Cliquez sur OK.


8. Sous le tableau, cliquez trois fois sur le bouton [+] afin d’ajouter trois lignes.

9. Dans le tableau, cliquez sur le champ id de la première ligne et saisissez 1.

10. Cliquez sur le champ firstname de la première ligne, appuyez sur les touches Ctrl+Espace pour afficher la liste d’autocomplétion et sélectionnez la variable TalendDataGenerator.getFirstName() dans la liste.

11. Cliquez sur le champ lastname de la première ligne, appuyez sur les touches Ctrl+Espace pour afficher la liste d’autocomplétion et sélectionnez la variable TalendDataGenerator.getLastName() dans la liste.

12. Répétez l’opération pour les lignes suivantes, afin d’obtenir les mêmes paramètres que dans la capture d’écran.

13. Double-cliquez sur le tMysqlOutput pour afficher sa vue Basic settings.

![tMysqlOutput.png](attachment:image.png)

15. Dans le champ **Table**, saisissez le nom de la table dans laquelle les données vont être chargées, par exemple : **users**.

16. Dans la liste **Action on table**, sélectionnez **Drop table if exists and create**, puis sélectionnez **Insert** dans la liste **Action on data**.

17. Cliquez sur **Sync columns** pour récupérer le schéma provenant du composant précédent.

**Créer le second sous-job**

Les composants suivants sont utilisés comme dans la première capture d’écran :

- Un **tRESTRequest** et un **tRESTResponse** de la famille **ESB > REST**,
- un **tFlowToIterate** de la famille **Orchestration**,
- un **tMysqlInput** de la famille **Databases > Mysql**,
- un **tXMLMap** de la famille **Processing**.

**Configurer le composant tRESTRequest**

**Pourquoi et quand exécuter cette tâche**

Pour ce faire, procédez comme suit :

**Procédure**

1. Double-cliquez sur le **tRESTRequest** dans l’espace de modélisation graphique afin d’afficher sa vue

![Basic settings](image)

**Basic settings.**

2. Renseignez le champ **REST Endpoint** avec l’emplacement de l’URI où le service Web REST-ful sera accessible pour les requêtes. Par exemple, "http://localhost:8088/users".
3. Cliquez sur le bouton [+ ] pour ajouter une ligne dans le tableau REST API Mapping.

4. Sélectionnez la nouvelle ligne et cliquez sur le bouton […] dans la colonne Output Flow afin d’ajouter un schéma au flux de sortie.

5. Dans la boîte de dialogue, nommez le flux de sortie getUsers. Un éditeur de schéma s’ouvre.

6. Dans l’éditeur de schéma, cliquez deux fois sur le bouton [+ ] pour ajouter deux lignes et paramétrez-les comme dans la capture d’écran ci-dessus.

7. Cliquez sur OK.

8. Retournez dans les propriétés du tRESTRequest, dans sa vue Basic settings, sélectionnez GET dans la liste de la colonne HTTP Verb.

9. Laissez la colonne URI Pattern telle qu’elle est.

10. Vous avez créé le flux de sortie du tRESTRequest, vous pouvez donc utiliser le lien correspondant pour le connecter au composant suivant : connectez le tRESTRequest au tFlowToIterate à l’aide d’un lien Row > getUsers.

11. Laissez les paramètres du tFlowToIterate tels qu’ils sont.

12. Connectez le tFlowToIterate au tMysqlInput à l’aide d’un lien Row > Iterate.

Configurer le composant tMysqlInput

Pourquoi et quand exécuter cette tâche

Pour ce faire, procédez comme suit :

Procédure

1. Double-cliquez sur le tMysqlInput pour afficher sa vue Basic settings.
2. Dans la liste **Property Type**, sélectionnez **Built-in** et renseignez les champs **Host**, **Port**, **Database**, **Username** et **Password** manuellement. Si vous avez centralisé vos informations de connexion dans le nœud **Metadata > DB Connections** du **Repository**, vous pouvez sélectionner **Repository** dans la liste. Les champs seront automatiquement renseignés.

Pour plus d’informations concernant le stockage des métadonnées, consultez le Guide utilisateur du **Studio Talend**.

3. Dans la liste **Schema**, sélectionnez **Built-in** et cliquez sur le bouton [...] à côté du champ **Edit schema**.

4. Dans l'éditeur de schéma, définissez le schéma exactement comme celui du **tFixedFlowInput**.

5. Dans le champ **Table Name**, renseignez le nom de la table dans laquelle les données sont stockées : **users**.

6. Dans la liste **Query Type**, sélectionnez **Built-in** et renseignez le champ **Query** avec la requête SQL suivante, permettant d’explorer les données de la base de données via la requête d’URI définie dans le composant **tRESTRequest**

   "select * from users where id >= " + globalMap.get("getUsers.from") + " and id <= " + globalMap.get("getUsers.to")"

**Configurer le composant tXMLMap**

**Procédure**

1. Cliquez-droit sur le composant **tMysqlInput**, maintenez et déposez sur le **tXMLMap** afin de connecter les deux composants.
2. Double-cliquez sur le **tXMLMap** dans l’espace de modélisation graphique pour ouvrir le **Map Editor**.

3. Cliquez sur le bouton [+] dans le coin supérieur droit afin d’ajouter une sortie. Nommez-la **ResponseUsers**.

4. Cliquez sur le bouton [+] en bas à droite pour ajouter deux colonnes de sortie.
   - Nommez la première colonne **body** et paramétrez son **Type** à **Document**.
   - Nommez la seconde colonne **string** et paramétrez son **Type** à **String**.

5. Cliquez-droit sur le nœud **root**, sélectionnez **Rename** dans la liste et renommez-le **users**.

6. Cliquez-droit sur le nœud **users** et sélectionnez **Create Sub-Element** afin de créer un sous-élément.
   - Nommez le sous-élément **user** dans la boîte de dialogue qui apparaît.

7. Cliquez-droit sur le nœud **user** créé à l’étape précédente et sélectionnez **As loop element**.

8. Sélectionnez la colonne **id** de la table **row2**, à gauche, et déposez-la sur le nœud **user** de la table **ResponseUsers**, à droite.

9. Dans la boîte de dialogue **[Selection]**, sélectionnez l’option **Create as attribute of target** puis cliquez sur **OK**.

10. Sélectionnez les colonnes **firstname** et **lastname** de la table **row2**et déposez-les sur le nœud **user** de la table **ResponseUsers**.
11. Dans la boîte de dialogue [Selection], sélectionnez l’option Create as sub-element of target node et cliquez sur OK.

12. Cliquez sur l’icône de clé anglaise, en haut de la table ResponseUsers pour ouvrir le panneau de paramétrage.

13. Paramétrez la fonctionnalité All in one à true. Ainsi, toutes les données XML sont écrites dans un seul flux.

14. Cliquez sur OK afin de sauvegarder les paramètres.

**Configurer le composant tRESTResponse**

**Procédure**

1. Connectez le tXMLMap au tRESTResponse à l’aide d’un lien Row > ResponseUsers.
2. Le schéma défini dans le tXMLMap est automatiquement récupéré par le tRESTResponse. Laissez les autres paramètres tels qu’ils sont.
Connecter les deux sous-jobs

Pourquoi et quand exécuter cette tâche
Les deux sous-jobs sont créés, vous pouvez donc les connecter.

Procédure
2. Sélectionnez `Trigger > OnSubjobOk` dans la liste.
3. Cliquez sur le composant `tRESTRequest` du second sous-job.

Résultats
Ainsi, lors de l’exécution du Job, le second sous-job est exécuté uniquement si l’exécution du premier s’est déroulée correctement.

Sauvegarder et exécuter le Job

Procédure
2. Ouvrez votre navigateur si vous souhaitez tester le service.
   Par exemple, utilisez la requête d’URI `?to=2` afin de récupérer les données des deux premiers utilisateurs.
La requête HTTP pour l'ID d'un utilisateur est acceptée par le service REST et la réponse HTTP est retournée au serveur.

Pour un cas d'utilisation appelant ce service REST à l'aide de composants Talend Open Studio for ESB, consultez Scénario 1 : Obtenir des informations sur un utilisateur en interagissant avec un service RESTful à la page 3385.

Scénario 3 : Service REST acceptant des requêtes HTTP POST

Ce scénario s'applique uniquement aux solutions Talend avec ESB.

Ce scénario décrit un Job composé de deux sous-jobs : le premier sous-job expose un service REST acceptant des requêtes HTTP POST de clients REST, écrivant des données dans une base de données après réception d'une requête HTTP et affichant les informations de l'échange serveur-client dans la console de la vue Run. Le second sous-job affiche les résultats des mises à jour de la base de données.
Construire le Job

Procédure

1. Créez un Job et ajoutez les composants suivants au Job en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la Palette : un tRESTRequest, un tMysqlOutput, un tXMLMap et un tRESTResponse.

2. Connectez le tRESTRequest au composant tMysqlOutput à l'aide d'un lien Row > Main et renommez le flux de sortie request dans cet exemple.

3. Reliez le composant tMysqlOutput au tXMLMap à l'aide d'un lien Row > Main.

4. Connectez le composant tXMLMap au tRESTResponse à l'aide d'un lien Row > Main que vous renommez response, par exemple. Cliquez sur OK dans la boîte de dialogue qui s'ouvre afin d'accepter la propagation du schéma du tRESTResponse.

5. Ajoutez un tMysqlInput et un tLogRow au Job en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la Palette. Reliez le tMysqlInput au tLogRow à l'aide d'un lien Row > Main.

6. Connectez le tMysqlOutput au tMysqlInput à l'aide d'un lien Trigger > OnComponentOk afin de relier les deux sous-jobs.

Configurer les composants

Configurer les paramètres de la requête REST

Procédure

1. Double-cliquez sur le composant tRESTRequest pour ouvrir sa vue Basic settings.

3. Cliquez sur le nom du flux de sortie, **request** dans cet exemple, dans la colonne **Output Flow** de la table **REST API Mapping** pour afficher le bouton [...]. Cliquez sur ce bouton pour ouvrir la boîte de dialogue [Schema]. Définissez comme suit le schéma pour le flux request. Ajoutez trois colonnes :
   - *id*, de type **Integer**, de longueur (Length) 2. Cochez la case **Key** pour en faire la colonne clé,
   - *first_name*, de type **String**,
   - *last_name*, de type **String**.

Cela fait, cliquez sur **OK** afin de valider les paramètres du schéma et fermer la boîte de dialogue.

4. Cliquez sur la colonne **HTTP Verb** et sélectionnez **POST** dans la liste.

5. Dans la colonne **URI Pattern**, saisissez le modèle d’URI des requêtes POST, "/post/{id}/{first_name}/{last_name}" , dans cet exemple.

Laissez les autres paramètres tels qu’ils sont.

6. Cliquez sur l’onglet **Advanced settings** et cochez la case **Log messages**.

**Configurer les paramètres de la base de données et de la réponse**

**Procédure**

1. Double-cliquez sur le composant **tMysqlOutput** pour ouvrir sa vue **Basic settings**.
2. Dans la liste Property Type, laissez Built-in sélectionné et renseignez manuellement les détails de connexion à la base de données dans les champs Host, Port, Database, Username et Password. Si vous avez centralisé vos informations de connexion à la base de données dans le nœud Metadata > DB Connections du Repository, vous pouvez choisir Repository dans la liste Property Type et sélectionner la connexion centralisée afin de renseigner automatiquement les champs. Pour plus d’informations concernant le stockage des métadonnées, consultez le Guide utilisateur du Studio Talend.

3. Renseignez le champ Table avec le nom de la table cible de la base de données, users dans cet exemple.

4. Sélectionnez les actions à effectuer sur la table de la base de données et sur les données selon vos besoins, dans les listes Action on table et Action on data, respectivement. Dans cet exemple, la table cible est créée si elle n’existe pas dans la base de données spécifiée. Les données provenant des requêtes clients sont insérées, ou mises à jour si elles existent déjà, dans la table de la base de données.

5. Cliquez sur le bouton Sync columns afin de synchroniser le schéma de la table et le schéma d’entrée.

6. Double-cliquez sur le composant tXMLMap pour ouvrir son éditeur de mapping.

7. Renommez le nœud root dans la table de sortie : cliquez-droit sur le nœud, sélectionnez Rename dans le menu contextuel et spécifiez un nouveau nom dans la boîte de dialogue qui s’ouvre, user dans cet exemple.

8. Sélectionnez les trois colonnes de la table d’entrée et déposez-les sur le nœud user, puis sélectionnez l’option Create as sub-element of target node dans la boîte de dialogue afin de configurer ces colonnes comme des sous-éléments du nœud user. Cela fait, cliquez sur OK afin de valider les mappings et fermer l’éditeur.
9. Double-cliquez sur le composant **tRESTResponse** et configurez les paramètres de réponse selon vos besoins. Dans cet exemple, laissez les paramètres par défaut.

**Configurer l’affichage des résultats**

**Procédure**

1. Double-cliquez sur le **tMysqlInput** pour ouvrir sa vue **Basic settings**.
2. Spécifiez les informations de connexion à la base de données, le nom de la table ainsi que le schéma de la table. Ces informations sont les mêmes que dans le composant tMysqlOutput.

3. Cliquez sur le bouton Guess Query afin de renseigner le champ Query avec l'instruction de requête.

4. Double-cliquez sur le tLogRow et sélectionnez l'option Table afin d’afficher le contenu de la base de données sous forme de tableau.

Sauvegarder et exécuter le Job

Procédure

1. Appuyez sur les touches Ctrl+S afin de sauvegarder votre Job.

2. Cliquez sur le bouton Run dans l'onglet Run ou appuyez sur la touche F6 pour l'exécuter.

La console affiche les informations relatives à l’implémentation du service, notamment son URL.

Lorsqu’une requête HTTP POST est reçue d’un client, la console affiche les informations relatives à l’échange, ainsi que le résultat de la mise à jour de la base de données.
Pour un cas d’utilisation appelant un service REST à l’aide de la méthode HTTP POST, consultez Scénario 2 : Mettre à jour les informations des utilisateurs via une interaction avec un service RESTful à la page 3392.

**Scénario 4 : Service REST acceptant les requêtes POST HTTP et envoyant des réponses**

Ce scénario s’applique uniquement aux solutions Talend avec ESB.

Ce scénario décrit comment construire un service REST acceptant les requêtes POST HTTP de type Document provenant d’un client et envoyant une réponse. Pour ce faire, deux Jobs sont construits : un pour créer le service REST et l’autre pour l’appeler.

**Créer le Job du service REST**

Glissez-déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un tRESTRequest, un tXMLMap et un tRESTResponse.
Configurer le composant tRESTRequest

Procédure

1. Double-cliquez sur le composant tRESTRequest dans l'espace de modélisation graphique pour afficher sa vue Basic settings.


3. Cliquez sur le bouton [+ ] pour ajouter une ligne à la table REST API Mapping.


   L'éditeur du schéma s'ouvre. Cliquez sur le bouton [+ ] pour ajouter une ligne body de type Document et cliquez sur OK pour sauvegarder le schéma.
5. De retour dans la vue **Basic settings** du composant, sélectionnez **POST** dans la liste de la colonne **HTTP Verb**.

6. Renseignez le champ dans la colonne **URI Pattern** en saisissant "/UserInfo". Laissez les valeurs par défaut des autres options.

**Configurer le corps de la réponse**

**Procédure**

1. Reliez le **tRESTRequest** au **tXMLMap** à l’aide d’un lien **Row > UserInfo**.
2. Double-cliquez sur le **tXMLMap** dans l’espace de modélisation graphique pour ouvrir son **Map Editor**.
3. Cliquez-droit sur le nœud root de la table d’entrée et sélectionnez Create Sub-Element. Nommez le sous-élément person dans la boîte de dialogue qui s’ouvre.

4. Cliquez-droit sur le nœud person créé dans l’étape précédente et sélectionnez As loop element.

5. Créez deux sous-éléments au nœud person en sélectionnant Create Sub-Element dans le menu contextuel. Nommez ces sous-éléments id et name.

6. Cliquez sur le bouton [+] dans le coin supérieur droit pour ajouter une sortie, que vous nommez Response.

7. Cliquez sur le bouton [+] en bas à droite pour ajouter une colonne body de type Document pour la sortie.


9. Cliquez-droit sur le nœud person créé dans l’étape précédente et sélectionnez As loop element.

10. Créez trois sous-éléments au nœud person en sélectionnant Create Sub-Element dans le menu contextuel, respectivement id, name et company dans cet exemple.

11. Sélectionnez la colonne id de la table d’entrée et déposez-la dans le champ Expression du nœud id de la table de sortie.

12. Sélectionnez la colonne name de la table d’entrée et déposez-la dans le champ Expression du nœud name de la table de sortie.

Cliquez dans le champ Expression du nœud name de la table de sortie et cliquez sur le bouton [...] dans ce champ pour afficher l’assistant [Expression Builder].

Dans la zone Expression, saisissez StringHandling.UPCASE((UserInfo.body:/root/person/name)) afin de convertir le nom de l’utilisateur en majuscules. Cliquez sur OK pour fermer l’assistant.
Pour plus d’informations concernant l’utilisation de l’Expression Builder, consultez le Guide utilisateur du Studio Talend.

13. Dans le champ Expression du nœud company, saisissez “Talend”.
14. Cliquez sur OK pour sauvegarder les paramètres.

Configurer le composant tRESTResponse

Procédure

1. Reliez le tXMLMap à un tRESTResponse à l’aide d’un lien Row > Response. Le schéma défini dans le tXMLMap est récupéré automatiquement dans le tRESTResponse.

2. Sélectionnez OK(200) dans la liste Return status code.
3. Laissez les autres paramètres tels qu’ils sont.

Lancer le Service

Sauvegardez le Job et appuyez sur F6 pour l’exécuter. Le service est démarré.
Créer le Job consommateur

Glissez-déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un tFixedFlowInput, un tXMLMap, un tRESTClient et un tLogRow.

Configurer le composant tFixedFlowInput

Procédure

1. Double-cliquez sur le tFixedFlowInput dans l’espace de modélisation graphique pour afficher sa vue Basic settings.

2. Cliquez sur le bouton Edit schema pour afficher l’éditeur du schéma. Cliquez sur le bouton [+ ] pour ajouter deux colonnes de type String, id et name. Cliquez sur OK pour fermer l’éditeur du schéma.
3. Dans la vue **Basic settings** du composant, dans la zone **Mode**, sélectionnez l’option **Use Inline Content**.

4. Dans le champ **Content**, saisissez l’id et le nom (**name**) de trois utilisateurs, par exemple :

```
1;ford
2;smith
3;thomson
```

**Configurer le composant tXMLMap**

**Procédure**

1. Reliez le **tFixedFlowInput** au **tXMLMap** à l’aide d’un lien **Row > Main**.
2. Reliez le **tXMLMap** au **tRESTClient** à l’aide d’un lien **Row > Main** et nommez le flux de sortie **request** dans cet exemple.
3. Double-cliquez sur le **tXMLMap** dans l’espace de modélisation graphique afin d’ouvrir le **Map Editor**.

5. Cliquez-droit sur le nœud person créé dans l’étape précédente et sélectionnez As loop element.

6. Créez deux sous-éléments pour le nœud person en sélectionnant Create Sub-Element dans le menu contextuel, id et name dans cet exemple.

7. Sélectionnez la colonne id de la table d’entrée et déposez-la dans le champ Expression du nœud id de la table de sortie.

8. Sélectionnez la colonne name de la table d’entrée et déposez-la dans le champ Expression du nœud name dans la table de sortie.

9. Cliquez sur OK afin de sauvegarder les paramètres.

**Configurer le composant tRESTClient**

**Procédure**

1. Double-cliquez sur le composant tRESTClient pour ouvrir sa vue Basic settings.

3. Renseignez le champ **Relative Path** avec “UserInfo”.

4. Dans la liste **HTTP Method**, sélectionnez **POST**.

5. Laissez les autres paramètres tels qu’ils sont.

6. Reliez le **tRESTClient** au **tLogRow** à l’aide d’un lien **Row > Response**. Laissez les autres paramètres par défaut dans le **tLogRow** afin de moniter les échanges de messages.

**Sauvegarder et exécuter le Job**

Sauvegarder le Job et appuyez sur **F6** pour l’exécuter. Les informations de l’utilisateur sont affichées dans la console, notamment **id, name et company**. Le nom de l’utilisateur a été converti en majuscules.
Scénario 5 : Service REST acceptant des requêtes POST HTTP dans un formulaire HTML

Ce scénario s’applique uniquement aux solutions Talend avec ESB.

Ce scénario décrit comment charger des données dans un service REST à l’aide de POST HTTP dans un fichier formulaire HTML. Pour ce faire, un Job est créé pour accepter les requêtes POST HTTP à l’aide du composant tRESTRequest.

Un formulaire HTML prédéfini UploadFile.html est utilisé afin de charger des données dans le service REST :

```html
<form action="http://localhost:8088/UserInfo" method="post" enctype="multipart/form-data">
  <input type="file" name="id" />
  <button>upload</button>
</form>
```

Le fichier customer.xml contenant l’ID d’un utilisateur va être chargé :

```xml
<customer>
  <id>100</id>
</customer>
```

Configurer le Job

Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un tRESTRequest, un tJavaRow, un tXMLMap et un tRESTResponse.
Configurer le composant tRESTRequest

Procédure

1. Double-cliquez sur le tRESTRequest dans l’espace de modélisation graphique pour afficher sa vue Basic settings.


3. Cliquez sur le bouton [+] pour ajouter une ligne dans la table REST API Mapping.


L’éditeur du schéma s’ouvre. Cliquez sur le bouton [+ ] pour ajouter une ligne id de type byte[] et saisissez multipart dans le champ Comment. Cliquez sur OK afin de sauvegarder le schéma.

5. Dans la vue Basic settings du composant, sélectionnez POST dans la liste de la colonne HTTP Verb.

6. Renseignez le champ de la colonne URI Pattern avec "/UserInfo".

7. Sélectionnez Multipart dans la liste Consumes. Laissez les paramètres par défaut pour les autres options.
Configurer le composant tJavaRow

Procédure
1. Reliez le tRESTRequest au tJavaRow à l’aide d’un lien Row > UserInfo.
2. Double-cliquez sur le tJavaRow dans l’espace de modélisation graphique pour afficher sa vue Basic settings.


```java
String result = new String(input_row.id, 0, input_row.id.length);

routines.system.Document doc = new routines.system.Document();

doc.setDocument(org.dom4j.DocumentHelper.parseText(result));

output_row.id = doc;

System.out.println(result);
```

Configurer le composant tXMLMap

Procédure
1. Connectez le tJavaRow au tXMLMap à l’aide d’un lien Row > Main.
2. Double-cliquez sur le tXMLMap dans l’espace de modélisation graphique pour ouvrir le Map Editor.
3. Cliquez sur le bouton [+] en bas à gauche pour ajouter une colonne id de type Document pour l’entrée.
4. Cliquez-droit sur le nœud root de la table d’entrée, sélectionnez Rename dans le menu contextuel et renommez-le customer.
5. Créez un sous-élément pour le nœud customer en sélectionnant Create Sub-Element dans le menu contextuel. Nommez le sous-élément id dans la boîte de dialogue qui s’ouvre.
6. Cliquez sur le bouton [+] en haut à droite pour ajouter une sortie et nommez-la response.
7. Cliquez sur le bouton [+] en bas à droite pour ajouter une colonne body de type Document pour la sortie.
8. Cliquez-droit sur le nœud root de la table d’entrée et sélectionnez Rename dans le menu contextuel et renommez-le customer.
10. Sélectionnez la colonne id de la table d’entrée et déposez-la sur le champ Expression du nœud id de la table de sortie.
11. Cliquez sur OK afin de sauvegarder les paramètres.

Configurer le composant tRESTResponse

Procédure
1. Reliez le tXMLMap au tRESTResponse à l’aide d’un lien Row > response.
   Le schéma défini dans le tXMLMap est récupéré automatiquement dans le tRESTResponse.
2. Sélectionnez **OK (200)** dans la liste **Return status code**.
3. Laissez les autres paramètres tels qu’ils sont.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Sauvegardez le Job et appuyez sur la touche **F6** pour l’exécuter.


3. Cliquez sur le bouton **Browse** pour naviguer jusqu’au fichier **customer.xml** contenant l’ID de l’utilisateur, puis cliquez sur **upload**.
4. La requête POST HTTP est acceptée par le service REST et l'ID de l'utilisateur est renvoyé au serveur.

Vous pouvez également voir la réponse dans la page source.

5. Le contenu du fichier d'entrée est affiché dans la console d'exécution.
[statistics] connecting to socket on port 3005
[statistics] connected
INFO: Setting the server's publish address to be http://localhost:8080/
>SelectChannelConnector@localhost:8080
<?xml version="1.0" encoding="ISO-8859-15"?>

<customer>
  <id>100</id>
</customer>
tRESTResponse

Ce composant retourne un code de statut HTTP spécifique au client comme réponse à une requête HTTP et/ou HTTPS.

**Remarque :**

Pour activer le support de HTTPS, vous devez générer un keystore et ajouter quelques propriétés liées à la configuration de sécurité HTTPS dans le fichier `org.ops4j.pax.web.cfg` de votre conteneur Talend Runtime. Pour plus d'informations, consultez le *Talend ESB Container Administration Guide* (en anglais).

**Propriétés du tRESTResponse Standard**

Ces propriétés sont utilisées pour configurer le tRESTResponse s'exécutant dans le framework de Jobs Standard.

Le composant tRESTResponse Standard appartient à la famille ESB.

Ce composant est adapté pour une utilisation au sein de la perspective Mediation du *Studio Talend* et il requiert l'utilisation du nœud du Repository Service et des assistants de création de Services.

**Basic settings**

<table>
<thead>
<tr>
<th>Return Body Type</th>
<th>Sélectionnez dans la liste le type du corps de la réponse, entre Document, String et Byte[].</th>
</tr>
</thead>
</table>
| Return status code | Sélectionnez un code de statut dans la liste pour indiquer le statut de la requête.  
- **<<Custom>>** : Cette option vous permet de personnaliser le code de statut. Saisissez dans le champ le code de statut de votre choix.  
- **Bad Request (400)** : La syntaxe de la requête est mauvaise ou la requête est impossible à effectuer.  
- **Internal Server Error (500)** : Des conditions inattendues rencontrées par le serveur l'empêchent d'exécuter la requête.  
- **OK (200)** : La requête a été exécutée.  
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant.  
Le schéma est en mode Built-in, c’est-à-dire créé et stocké localement pour ce composant.  
Cliquez sur Edit schema pour modifier le schéma.  
Le flux d’entrée est récupéré du composant connecté au tRESTResponse. Toute modification dans le flux d’entrée du tRESTResponse est automatiquement propagée au composant précédent. |
Le flux de sortie du **tRESTResponse** est un champ codé en dur nommé **body**, dont le type est défini dans la liste **Return Body Type**.

**Advanced settings**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Response Headers</strong></td>
<td>Saisissez la(les) paire(s) nom-valeur pour passer des réponses supplémentaires concernant la réponse. Pour plus d’informations concernant les en-têtes de réponses, consultez le site <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html#sec6.2">http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html#sec6.2</a> (en anglais).</td>
</tr>
<tr>
<td><strong>Drop JSON Response Root</strong></td>
<td>Cochez cette case pour déposer l’élément JSON racine dans la réponse.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>NB_LINE</strong> : nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant permet d’exposer un Job <strong>Talend</strong> en tant que service et de transmettre une requête à un service dans un Job puis retourne les résultats du Job en tant que réponse du service.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Le composant <strong>tRESTResponse</strong> doit être utilisé avec le <strong>tRESTRequest</strong> afin de fournir les résultats d’un Job en tant que réponse, dans le cas d’une communication de style requête-réponse.</td>
</tr>
</tbody>
</table>
Scénario associé

Pour un scénario dans lequel le tRESTResponse est utilisé, consultez Scénario 1 : Service REST acceptant des requêtes HTTP GET et envoyant des réponses à la page 3404.
tRiakBucketList

Ce composant récupère une liste de buckets d’un cluster Riak et fait une boucle dessus.

Propriétés du tRiakBucketList Standard

Ces propriétés sont utilisées pour configurer le tRiakBucketList s’exécutant dans le framework de Jobs Standard.

Le composant tRiakBucketList Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Si vous cochez la case Use an existing connection, la table Nodes n’est pas disponible. |
| Nodes | Saisissez les informations requises pour une connexion au cluster Riak.

• *Host* : saisissez l’adresse IP ou le nom d’hôte d’un nœud Riak du cluster Riak auquel vous souhaitez vous connecter.

• *Port* : saisissez le numéro du port d’écoute du nœud Riak. Vous pouvez laisser ce champ vide afin d’utiliser sa valeur par défaut, 8098.

• *Riak path* : saisissez le chemin d’accès Riak au nœud Riak. Vous pouvez laisser ce champ vide afin d’utiliser sa valeur par défaut, *riak*.

L’URL d’accès au nœud Riak peut être, par exemple, *http://127.0.0.1:8098/riak*.

Pour plus d’informations sur les concepts liés à Riak, consultez la page http://docs.basho.com/riak/latest/theory/concepts/ (en anglais). |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

Global Variables

| Global Variables | CURRENT_BUCKET_NAME : nom du bucket courant. Cette variable est une variable *Flow* et retourne une chaîne de caractères. |
**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

---

**Utilisation**

| Règle d’utilisation | Ce composant est utilisable en tant que composant standalone. |

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tRiakClose

Ce composant ferme une connexion active à un cluster Riak afin de libérer les ressources occupées.

Propriétés du tRiakClose Standard

Ces propriétés sont utilisées pour configurer le tRiakClose s’exécutant dans le framework de Jobs Standard.

Le composant tRiakClose Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

| Component List | Sélectionnez la connexion active au cluster Riak que vous souhaitez fermer. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Riak, en particulier le tRiakConnection. |
Scénario associé

Pour un scénario utilisant le **tRiakClose**, consultez *Scénario : Exporter des données depuis un bucket Riak vers un fichier local* à la page 3451.
tRiakConnection

Ce composant ouvre et réutilise la connexion au cluster Riak créée.

Propriétés du tRiakConnection Standard

Ces propriétés sont utilisées pour configurer le tRiakConnection s’exécutant dans le framework de Jobs Standard.

Le composant tRiakConnection Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Saisissez les informations requises pour une connexion au cluster Riak.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- <strong>Host</strong>: saisissez l’adresse IP ou le nom d’hôte d’un nœud Riak du cluster Riak auquel vous souhaitez vous connecter.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Port</strong>: saisissez le numéro du port d’écoute du nœud Riak. Vous pouvez laisser ce champ vide afin d’utiliser sa valeur par défaut, 8098.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Riak path</strong>: saisissez le chemin d’accès Riak au nœud Riak. Vous pouvez laisser ce champ vide afin d’utiliser sa valeur par défaut, riak.</td>
</tr>
</tbody>
</table>


Pour plus d’informations sur les concepts liés à Riak, consultez la page http://docs.basho.com/riak/latest/theory/concepts/ (en anglais).

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace |

pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec d’autres composants, en particulier le <strong>tRiakClose</strong>.</th>
</tr>
</thead>
</table>

**Scénario associé**

Pour un scénario utilisant le **tRiakConnection**, consultez Scénario : Exporter des données depuis un bucket Riak vers un fichier local à la page 3451.
Ce composant extrait les données souhaitées d'un bucket d'un nœud Riak afin de les stocker ou d'y apporter des changements.

Le composant tRiakInput lit les données depuis un bucket Riak et les envoie dans le flux Talend.

**Propriétés du tRiakInput Standard**

Ces propriétés sont utilisées pour configurer le tRiakInput s'exécutant dans le framework de Jobs Standard.

Le composant tRiakInput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. 

Cliez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content. |
| --- | --- |
| Use existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie. 

Si vous cochez la case Use an existing connection, la table Nodes n’est pas disponible. |
| Nodes | Saisissez les informations requises pour une connexion au cluster Riak. 

- **Host** : saisissez l'adresse IP ou le nom d'hôte d'un nœud Riak du cluster Riak auquel vous souhaitez vous connecter. |
**Port** : saisissez le numéro du port d’écoute du nœud Riak. Vous pouvez laisser ce champ vide afin d’utiliser sa valeur par défaut, 8098.

**Riak path** : saisissez le chemin d’accès Riak au nœud Riak. Vous pouvez laisser ce champ vide afin d’utiliser sa valeur par défaut, riak.


<table>
<thead>
<tr>
<th>Bucket</th>
<th>Saisissez le nom du bucket dont vous souhaitez lire les données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Saisissez la clé associée aux données que vous souhaitez lire.</td>
</tr>
<tr>
<td>Output key to column</td>
<td>Cochez cette case et sélectionnez dans la liste la colonne vers laquelle sont envoyées les données.</td>
</tr>
<tr>
<td>Values column</td>
<td>Personnalisez les colonnes vers lesquelles les données sont envoyées.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <strong>After</strong> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
Scénario : Exporter des données depuis un bucket Riak vers un fichier local

Ce scénario s'applique uniquement aux solutions Talend avec Big Data.

Le scénario suivant permet de créer un Job lisant des données depuis un bucket Riak et les écrivant dans un fichier txt local.

Prérequis : Le bucket Riak depuis lequel vous souhaitez exporter des données doit exister. Dans cet exemple, les données du bucket computer sont exportées et ce bucket a déjà importé les données suivantes :

<table>
<thead>
<tr>
<th>id</th>
<th>company</th>
<th>brand</th>
<th>price</th>
<th>owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Dell</td>
<td>Inspiron 15</td>
<td>299</td>
<td>Amanda</td>
</tr>
<tr>
<td>002</td>
<td>Dell</td>
<td>Inspiron 15R</td>
<td>549</td>
<td>Linda</td>
</tr>
<tr>
<td>003</td>
<td>HP</td>
<td>Pavilion 500-210qe</td>
<td>539</td>
<td>Marina</td>
</tr>
<tr>
<td>004</td>
<td>HP</td>
<td>Pavilion 500-075</td>
<td>599</td>
<td>Diana</td>
</tr>
</tbody>
</table>

Déposer et relier les composants

Procédure

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : un tRiakConnection, un tRiakInput, un tFileOutputDelimited et un tRiakClose.
2. Reliez le tRiakConnection au tRiakInput à l'aide d'un lien Trigger > On Subjob Ok.
3. Reliez le tRiakInput au tFileOutputDelimited à l'aide d'un lien Row > Main.
4. Reliez le tFileOutputDelimited au tRiakClose à l'aide d'un lien Trigger > On Component Ok.

Configurer les composants

Ouvrir une connexion à Riak

Procédure

1. Double-cliquez sur le tRiakConnection afin d'ouvrir sa vue Basic settings dans l'onglet Component.
2. Dans la table **Nodes**, saisissez les informations du cluster Riak auquel vous souhaitez vous connecter.

**Exporter les données depuis un bucket Riak vers un fichier local**

**Procédure**

1. Double-cliquez sur le **tRiakInput** afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Cliquez sur **Edit schema** afin de définir la structure des données exportées. Dans cet exemple, trois colonnes sont définies : *id*, *company* et *price*. 
3. Cochez la case **Use an existing connection** puis sélectionnez la connexion que vous avez configurée précédemment, *tRiakConnection_1* dans cet exemple.

4. Dans le champ **Bucket**, saisissez un nom pour le bucket à partir duquel vous souhaitez exporter les données, *computer* dans cet exemple.

5. Cochez la case **Output key to column** puis sélectionnez la colonne souhaitée dans la liste, *id* dans cet exemple.

6. Dans la table **Value columns**, cliquez deux fois sur le bouton et sélectionnez *company* et *price* dans la liste respectivement.

**Écrire les données dans un fichier local**

**Procédure**

1. Double-cliquez sur le *tFileOutputDelimited* afin d'ouvrir sa vue **Basic settings** dans l'onglet **Component**.

2. Dans le champ **File Name**, saisissez le chemin complet vers le fichier local dans lequel vous souhaitez stocker les données exportées, *D:/Output/computer.txt* dans cet exemple.

3. Cochez la case **Include Header**.

4. Laissez les autres paramètres tels qu’ils sont.
**Fermer la connexion à Riak**

**Procédure**

1. Double-cliquez sur le `tRiakClose` afin d’ouvrir sa vue **Basic settings** dans l’onglet **Component**.

![Component List](image)

2. Dans la liste **Component List**, sélectionnez la connexion que vous souhaitez fermer, `tRiakConnection_1` dans cet exemple.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** dans l’onglet **Run** afin d’exécuter le Job.
3. Parcourez votre système vers le dossier local contenant le fichier et vérifiez qu’il contient les données exportées depuis le bucket Riak.

![Notepad](image)
tRiakKeyList

Ce composant récupère une liste de clés dans un bucket Riak et fait une boucle dessus afin de procéder à des analyses et à des développements.

**Propriétés du tRiakKeyList Standard**

Ces propriétés sont utilisées pour configurer le tRiakKeyList s’exécutant dans le framework de Jobs Standard.

Le composant tRiakKeyList Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. Si vous cochez la case Use an existing connection, la table Nodes n’est plus disponible.</th>
</tr>
</thead>
</table>
| Nodes                       | Saisissez les informations requises pour une connexion au cluster Riak.  
• **Host** : saisissez l’adresse IP ou le nom d’hôte d’un nœud Riak du cluster Riak auquel vous souhaitez vous connecter.  
• **Port** : saisissez le numéro du port d’écoute du nœud Riak. Vous pouvez laisser ce champ vide afin d’utiliser sa valeur par défaut, 8098.  
• **Riak path** : saisissez le chemin d’accès Riak au nœud Riak. Vous pouvez laisser ce champ vide afin d’utiliser sa valeur par défaut, riak.  
Pour plus d’informations sur les concepts liés à Riak, consultez la page http://docs.basho.com/riak/latest/theory/concepts/ (en anglais). |
| Bucket                      | Saisissez le nom du bucket dont vous souhaitez récupérer les clés. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |
Global Variables

CURRENT_KEY : clé courante. Cette variable est une variable Flow et retourne une chaîne de caractères.

NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d'utilisation

Ce composant est utilisable en tant que composant standalone.

Scénario associé

Aucun scénario n'est disponible pour la version Standard de ce composant.
tRiakOutput

Ce composant reçoit des données du composant précédent, écrit ou efface des données dans un bucket d'un cluster Riak.

Propriétés du tRiakOutput Standard

Ces propriétés sont utilisées pour configurer le tRiakOutput s'exécutant dans le framework de Jobs Standard.

Le composant tRiakOutput Standard appartient aux familles Big Data et Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Basic settings

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |

| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.
Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma |
| Use existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Si vous cochez la case **Use an existing connection**, la table **Nodes** n’est pas disponible. |
| Nodes | Saisissez les informations requises pour une connexion au cluster Riak.

- **Host** : saisissez l'adresse IP ou le nom d'hôte d'un nœud Riak du cluster Riak auquel vous souhaitez vous connecter.
- **Port** : saisissez le numéro du port d’écoute du nœud Riak. Vous pouvez laisser ce champ vide afin d’utiliser sa valeur par défaut, 8098.
- **Riak path** : saisissez le chemin d’accès Riak au nœud Riak. Vous pouvez laisser ce champ vide afin d’utiliser sa valeur par défaut, riak.


Pour plus d’informations sur les concepts liés à Riak, consultez la page http://docs.basho.com/riak/latest/theory/concepts/ (en anglais). |
| Bucket | Spécifiez le nom du bucket contenant les données sur lesquelles apporter des changements. |
| Action on data | Dans le bucket spécifié, vous pouvez effectuer les actions suivantes :

- **Upsert** : Insère les données si elles n’existent pas ou met à jour les données existantes.
- **Delete** : Supprime les valeurs associées avec la clé spécifiée.

Si vous sélectionnez **Delete** dans la liste **Action on data**, seule la liste **Key column** est disponible dans la zone **Key**. |
| Auto generate the key | Cochez cette case pour laisser le système Riak générer automatiquement des clés pour les valeurs. |
| Key column | Sélectionnez une colonne dans la liste afin d’écrire les données qu’elle contient dans un bucket Riak en tant que clés. Notez que la clé doit être unique dans le bucket. |
| Value columns | Personnalisez les colonnes afin d’écrire leurs données dans un bucket Riak en tant que valeurs. |
Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

Global Variables

| Global Variables | Variables correspondant à l’utilisation des sorties 
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>UPSER_T_NB_LINE</strong> : nombre de lignes insérées et mises à jour. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td><strong>DELETE_NB_LINE</strong> : nombre de lignes supprimées. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

Utilisation

| Règle d’utilisation | Ce composant est utilisé en tant que composant de sortie et nécessite un lien d’entrée. |

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tRouteFault

Ce composant envoie des messages d'un Job d'intégration de données à une Route de médiation et marque les messages comme Fault (en erreur).

Propriétés du tRouteFault Standard

Ces propriétés sont utilisées pour configurer le tRouteFault s'exécutant dans le framework de Jobs Standard.

Le composant tRouteFault Standard appartient à la famille ESB.

Ce composant est adapté pour une utilisation au sein de la perspective Mediation du Studio Talend. Il requiert l'utilisation du nœud du Repository Service et des assistants de création de Services.

Basic settings

| Output Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

  • View schema : sélectionnez cette option afin de voir le schéma.

  • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

  • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

  Cliquez sur Sync columns afin de récupérer automatiquement les colonnes du composant précédent. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Le schéma sera créé et conservé pour ce composant seulement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé.</td>
</tr>
<tr>
<td>Camel</td>
<td>Saisissez les paramètres de message correspondant aux colonnes définies dans le schéma via le bouton Edit schema.</td>
</tr>
<tr>
<td>Type</td>
<td>Choisissez le type :</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Body</td>
<td>Utilisé pour configurer le corps d'une Route, équivaut au code <code>exchange.setOutput().setBody(...)</code></td>
</tr>
<tr>
<td>Property</td>
<td>Utilisé pour configurer la propriété d'une Route, équivaut au code <code>exchange.setProperty(name, value)</code></td>
</tr>
<tr>
<td>System</td>
<td>Utiliser pour configurer la propriété du système, équivaut au code <code>System.setProperty(name, value)</code></td>
</tr>
<tr>
<td>Header</td>
<td>Utilisé pour configurer l’en-tête de la sortie, équivaut au code <code>exchange.setOutput().setHeader(name, value)</code></td>
</tr>
</tbody>
</table>

| Name               | Cette colonne est utilisée pour configurer le nom du paramètre lorsque son type est `Property`, `System` ou `Header`. Cette colonne est ignorée lorsque son type est `Body`. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant est utilisé comme composant de fin et nécessite un composant d’entrée. |
| Limitation          | Un Job utilisant un `tRouteFault` doit être lancé dans une Route comprenant le composant `cTalendJob`. |
Scénario : Échanger des messages entre un Job et une Route

Ce scénario s'applique uniquement aux solutions Talend avec ESB.

Ce scénario décrit comment échanger des messages entre un Job et une Route à l'aide des composants tRouteInput et tRouteFault. Pour ce faire, un Job est créé pour recevoir un message d'une Route et marquer le message comme Fault, en erreur. Une Route est construite pour envoyer le message au Job et gérer le message en erreur.

Créer un Job d'intégration de données

Dans cette section, un Job nommé RouteCommunication_tRouteFault est construit.

Construire le flux du message

Pourquoi et quand exécuter cette tâche

![Diagramme du flux de message]

Procédure

1. Glissez-déposez un tRouteInput et un tRouteFault de la Palette dans l'espace de modélisation graphique.
2. Cliquez-droit sur le tRouteInput, sélectionnez Row > Main dans le menu contextuel et cliquez sur le composant tRouteFault.

Configurer le traitement du message

Procédure

1. Double-cliquez sur le tRouteInput pour ouvrir sa vue Basic settings.

![Vue Basic settings du tRouteInput]

2. Cliquez sur le bouton [...] à côté du champ Edit Schema. Dans la boîte de dialogue du schéma, cliquez sur le bouton [+] pour ajouter une ligne de type String, que vous nommez body. Cliquez sur OK pour fermer la boîte de dialogue.
3. Dans le champ **Simple Expression** de l’élément *body*, saisissez "$\{in.body\}" pour obtenir le corps du message d’entrée de la Route.

4. Double-cliquez sur le composant tRouteFault pour afficher sa vue **Basic settings**.

5. Assurez-vous que le tRouteFault a bien un élément *body*. Configurez son type à **Body**.

6. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.

**Créer une Route de médiation**

Dans cette section, vous allez créer une Route pour envoyer un message au Job et gérer le message en erreur.
Glisser et relier les composants

Pourquoi et quand exécuter cette tâche

![Diagramme de modélisation graphique]

1. Glissez-déposez un cTimer, un cSetBody, un cJavaDSLProcessor, un cTalendJob, un cOnException et un cProcessor de la Palette dans l’espace de modélisation graphique.
2. Reliez les composants à l’aide de liens Row > Route, comme dans la capture d’écran ci-dessus.
3. Renommez les composants afin de mieux identifier leur rôle au sein de la Route.

Configurer les composants

Procédure

1. Double-cliquez sur le cTimer pour ouvrir sa vue Basic settings.

![Vues Basic settings de cTimer et cSetBody]

2. Saisissez 1 dans le champ Repeat afin de déclencher une fois un échange de message.
3. Double clquez sur le cSetBody pour afficher sa vue Basic settings.
4. Sélectionnez Simple dans la liste Language et saisissez Hello! dans le champ Expression.
5. Double-cliquez sur le composant **cJavaDSLProcessor** pour ouvrir sa vue **Basic settings**.

6. Saisissez `.handleFault()` dans le champ **Code** pour que le message en erreur soit pris en compte comme une exception.

7. Double-cliquez sur le **cTalendJob** pour afficher sa vue **Basic settings**.

8. Sélectionnez **Repository**, puis choisissez le Job **RouteCommunication_tRouteFault** afin d’appeler le Job créé dans la section précédente.

9. Double-cliquez sur le composant **cOnException** pour afficher sa vue **Basic settings**.
10. Cliquez sur le bouton [+ ] pour ajouter une ligne à la table Exceptions et saisissez org.apache.camel.CamelException dans le champ Exception pour gérer les exceptions.

11. Double-cliquez sur le cProcessor pour afficher sa vue Basic settings.

12. Personnalisez le code dans le champ Code pour afficher le corps du message :

```java
System.out.println(exchange.getIn());
```

13. Appuyez sur les touches Ctrl+S afin de sauvegarder votre Route.

**Exécuter la Route**

Cliquez sur la vue Run pour l’afficher et cliquez sur le bouton Run pour démarrer l’exécution de la Route. Vous pouvez également appuyer sur F6 pour l’exécuter.

Le message Hello! est envoyé de la Route au Job et récupérée comme une exception :
[statistics] connecting to socket on port 3783
[statistics] connected

Message: Hello!


Processed by failure processor: FallbackErrorHandler[Channel[DelegateSync[local_project.routefault_0_1.RouteFault]@2511045b]]

---

Message History

----------------------------------------
<table>
<thead>
<tr>
<th>RouteId</th>
<th>ProcessorId</th>
<th>Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Starter_cTimer_1</td>
<td>[Starter_cTimer_1</td>
<td>]</td>
</tr>
<tr>
<td>[timer://cTimer_1?delay=1000&amp;repeatCount=1</td>
<td>40 ]</td>
<td></td>
</tr>
<tr>
<td>[Starter_cTimer_1</td>
<td>[cJavaDSLProcessor_</td>
<td>[setBody/simple{Hello!}]</td>
</tr>
<tr>
<td>] [20 ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Starter_cTimer_1</td>
<td>[cTalendJob_1</td>
<td>]</td>
</tr>
<tr>
<td>[talend://local_project.routecommunication_routefault_0_1.RouteCommunication_t</td>
<td>17</td>
<td>]</td>
</tr>
<tr>
<td>[cProcessor_1</td>
<td>[Processor@0x251104fb</td>
<td></td>
</tr>
<tr>
<td>] [0</td>
<td>]</td>
<td></td>
</tr>
</tbody>
</table>

---

Exchange

----------------------------------------
<table>
<thead>
<tr>
<th>Id</th>
<th>ExchangePattern</th>
<th>Headers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID-Talend-PC-7919-1437988941394-0-2</td>
<td>InOnly</td>
<td>{breed:crumbId=ID-Talend-PC-7919-1437988941394-0-1, CamelRedelivered=false, CamelRedeliveryCounter=0, firedTime=Mon Jul 27 17:22:23 CST 2015}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BodyType</th>
<th>Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Hello!</td>
</tr>
</tbody>
</table>

---

StackTrace

----------------------------------------------------------------------
crg.apache.camel.CamelException: Hello!
at crg.apache.camel.processor.interceptor.HandleFaultInterceptor.handleFault
| (HandleFaultInterceptor.java:75) | 3467 |
tRouteInput

Ce composant échange des messages entre un Job d'intégration de données et une Route de médiation.

Propriétés du tRouteInput Standard

Ces propriétés sont utilisées pour configurer le tRouteInput s'exécutant dans le framework de Jobs Standard.

Le composant tRouteInput Standard appartient à la famille ESB.

Ce composant est adapté pour une utilisation au sein de la perspective Mediation du Studio Talend. Il requiert l'utilisation du nœud du Repository Service et des assistants de création de Services.

Basic settings

| Input Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

| Built-in | Le schéma sera créé et conservé pour ce composant seulement. |

| Repository | Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. |

| Camel | Saisissez les paramètres du message correspondant aux colonnes définies dans le schéma via le bouton Edit schema à l'aide du langage Simple Expression.
Pour plus d'informations concernant le langage Simple Expression, consultez le site (en anglais) http://camel.apache.org/simple.html. |
Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est utilisé en tant que composant de début et doit avoir une sortie. Il est obligatoire d’utiliser un tRouteInput dans un Job d’intégration de données à appeler par une Route utilisant le composant cTalendJob. Cela empêche le Job référencé de démarrer automatiquement lorsqu’il est déployé dans Talend Runtime. Il est alors démarré uniquement lorsqu’il est appelé par la Route. Pour plus d’informations concernant l’utilisation du cTalendJob, consultez Talend Help Center (https://help.talend.com). |

| Limitation | Un Job utilisant un tRouteInput doit être lancé dans une Route utilisant le composant cTalendJob. |

Scénario : Échanger des messages entre un Job et une Route

Ce scénario s’applique uniquement aux solutions Talend avec ESB.

Ce scénario décrit comment accepter des messages provenant d’une Route à l’aide du composant tRouteInput et comment envoyer des messages à une Route à l’aide du composant tRouteOutput. Pour ce faire, vous devez construire un Job et une Route pour appeler le Job à l’aide du composant cTalendJob.

**Créer un Job d’intégration de données**

Dans cette section, vous allez construire un Job nommé RouteCommunication acceptant le message d’une Route, modifier le corps du message et renvoyer le message à la Route.

**Construire le flux du message**

**Pourquoi et quand exécuter cette tâche**

![Diagramme de flux de message](image_url)

**Procédure**

1. Glissez-déposez un tRouteInput, un tLogRow, un tReplace et un tRouteOutput de la Palette dans l’espace de modélisation graphique.
2. Reliez les composants à l’aide de liens Row > Main, comme dans l’image ci-dessus.

**Configurer le traitement du message**

**Procédure**

1. Double-cliquez sur le tRouteInput pour ouvrir sa vue Basic settings.

![Vue Basic settings](image_url)

3. Dans le champ Simple Expression de l’élément body, saisissez "${in.body}" pour obtenir le corps du message d’entrée provenant de la Route.

4. Le composant tLogRow permet de moniter les échanges de messages et ne nécessite aucune configuration.

5. Double-cliquez sur le composant tReplace pour ouvrir sa vue Basic settings.

6. Cliquez sur le bouton Sync columns pour récupérer le schéma du tLogRow. Dans la boîte de dialogue qui s’ouvre, cliquez sur Yes afin de propager le schéma au composant suivant.

   Cliquez sur le bouton [+ ] pour ajouter une ligne à la table Search/Replace. Saisissez World dans la colonne Search et Talend dans la colonne Replace with. Décrochez la case Whole word.

7. Double-cliquez sur le composant tRouteOutput pour afficher sa vue Basic settings dans l’onglet Component.
8. Assurez-vous que le tRouteOutput contient un élément *body*. Configurez son type à *Body*.
9. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job.

**Créer une Route de médiation**

Dans cette section, vous allez créer une Route envoyant un message au Job et recevant le message renvoyé à la Route.

**Déposer et relier les composants**

**Pourquoi et quand exécuter cette tâche**

![Diagramme de dépose et relier les composants](image)

**Procédure**

1. Glissez-déposez un cTimer, un cSetBody, un cTalendJob et un cLog de la Palette dans l’espace de modélisation graphique.
2. Reliez les composants à l’aide de liens **Row > Route** comme dans la capture d’écran.
3. Renommez les composants afin de mieux identifier leur rôle.

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le cTimer pour ouvrir sa vue **Basic settings**.
2. Dans le champ **Repeat**, saisissez 1 pour générer une fois l’échange de messages. Laissez les autres paramètres par défaut.

3. Double-cliquez sur le composant **cSetBody** pour afficher sa vue **Basic settings**.

4. Sélectionnez **Simple** dans la liste **Language** et saisissez *Hello World!* dans le champ **Expression**.

5. Double-cliquez sur le composant **cTalendJob** pour afficher sa vue **Basic settings**.

6. Sélectionnez le Job **RouteCommunication** du **Repository**.

7. Le composant **cLog** va monitorer les échanges de messages et ne nécessite aucune confirmation.

8. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Route.
**Exécuter la Route**

Cliquez sur la vue **Run** pour l’afficher puis sur le bouton **Run** pour démarrer l’exécution de votre Route. Vous pouvez également appuyer sur **F6** pour l’exécuter. Dans le journal d’exécution, vous pouvez constater que le message *Hello World!* est envoyé au Job. Le corps du message est changé en *Hello Talend!* et renvoyé à la Route.
tRouteOutput

Ce composant envoie des messages d'un Job d'intégration de données à une Route de médiation.

Propriétés du tRouteOutput Standard

Ces propriétés sont utilisées pour configurer le tRouteOutput s'exécutant dans le framework de Jobs Standard.

Le composant tRouteOutput Standard appartient à la famille ESB.

Ce composant est adapté pour une utilisation au sein de la perspective Mediation du Studio Talend. Il requiert l'utilisation du nœud du Repository Service et des assistants de création de Services.

Basic settings

| Output Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. Cliquez sur Sync columns afin de récupérer automatiquement les colonnes du composant précédent. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Le schéma sera créé et conservé pour ce composant seulement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé.</td>
</tr>
<tr>
<td>Camel</td>
<td>Saisissez les paramètres de message correspondant aux colonnes définies dans le schéma via le bouton Edit schema.</td>
</tr>
<tr>
<td>Type</td>
<td>Choisissez le type :</td>
</tr>
</tbody>
</table>

Camel
**tRouteOutput**

Body : Utilisé pour configurer le corps d'une Route, équivaut au code `exchange.getOutputStream().setBody(...)`.

Property : Utilisé pour configurer la propriété d'une Route, équivaut au code `exchange.setProperty(name, value)`.

System : Utiliser pour configurer la propriété du système, équivaut au code `System.setProperty(name, value)`.

Header : Utilisé pour configurer l'en-tête de la sortie, équivaut au code `exchange.getOutputStream().setHeader(name, value)`.

| Name | Cette colonne est utilisée pour configurer le nom du paramètre lorsque son type est Property, System ou Header. Cette colonne est ignorée lorsque son type est Body. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

| Global Variables | ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| Règle d'utilisation | Ce composant est utilisé comme composant de fin et nécessite un composant d'entrée. Il est obligatoire d'utiliser un tRouteInput dans le Job d'intégration de données devant être appelé par la Route à l'aide du cTalendJob. Cela empêche le Job référencé de démarrer automatiquement lorsqu'il est déployé dans Talend Runtime. Le Job démarre alors uniquement lorsqu'il est appelé par la Route. |
| Limitation | Un Job utilisant un `tRouteOutput` doit être lancé dans une Route comprenant un composant `cTalendJob`. |

**Scénario associé**

Pour un scénario utilisant le `tRouteOutput`, consultez *Scénario : Échanger des messages entre un Job et une Route* à la page 3469.
**tRowGenerator**

Ce composant est utilisé à des fins de test, pour créer un flux d’entrée dans le cadre de jeux de tests.

Le composant tRowGenerator génère autant de lignes et de champs que vous souhaitez alimenter par des valeurs prises de façon aléatoire dans une liste définie.

**Propriétés du tRowGenerator Standard**

Ces propriétés sont utilisées pour configurer le tRowGenerator s’exécutant dans le framework de Jobs Standard.

Le composant tRowGenerator Standard appartient à la famille Misc.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

  - **View schema**: sélectionnez cette option afin de voir le schéma.
  - **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
  - **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |

| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |

**RowGenerator editor**

L’éditeur vous permet de définir précisément les colonnes et la nature des données à générer. Vous pouvez prédéfinir des routines ou saisir vous-même la fonction à utiliser pour générer les données souhaitées.
Notez que, dans un Job Storm, la valeur -1 dans le champ **Number of rows for RowGenerator** du **RowGenerator editor** signifie la génération infinie de lignes de données d’entrée.

### Advanced settings

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

- **NB_LINE** : nombre de lignes traitées. Cette variable est une variable *After* et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

**Règle d’utilisation**

La simplicité d’utilisation de l’éditeur du **tRowGenerator** vous permet de générer des données de façon aléatoire sans connaissance du langage **Java**.

### Définition du schéma

Pour commencer, vous devez définir la structure des données à générer.

- **Ajoutez autant de colonnes que nécessaire à l’aide du bouton **plus** [+]**.
- **Dans la zone **Columns**, saisissez les noms des colonnes à créer et cochez la case **Key** si nécessaire.**
- **Assurez-vous d’avoir correctement défini la nature des données contenues dans la colonne en sélectionnant le type dans la colonne **Type**. En fonction du type sélectionné, la liste des fonctions proposées dans la colonne **Function** diffère. Cette information est obligatoire.**
• D’autres informations, bien que facultatives, peuvent être utiles, telles que les colonnes **Length**, **Precision** ou **Comment**. Mais vous avez la possibilité de cacher les colonnes que vous n’utilisez pas, en cliquant sur le bouton **Columns** dans la barre d’outils et en décochant les colonnes que vous souhaitez cacher.

• Dans la zone **Function**, sélectionnez la routine/fonction prédéfinie si l’une des propositions correspond à vos besoins. Vous pouvez également ajouter à cette liste toute routine que vous avez développée et stockée dans la partie **Routine** du **Repository**. Vous avez enfin la possibilité de saisir directement la fonction que vous souhaitez utiliser dans la vue **Function**. Voir également : **Définition d’une fonction** à la page 3480.

• Cliquez sur **Refresh** pour visualiser un aperçu des données générées.

• Saisissez le nombre de lignes à générer. Plus le nombre de lignes est élevé, plus la durée d’exécution sera longue pour cette opération de génération.

**Définition d’une fonction**

Vous avez sélectionné les trois points […] dans la colonne **Function** de l’éditeur du Schéma, car vous souhaitez personnaliser la fonction de génération.

• Sélectionnez l’onglet **Function parameters**.

• La zone **Parameter** affiche **Customized parameter** comme nom du paramètre (en lecture seule).

• Dans la zone **Value**, saisissez la fonction Java que vous souhaitez utiliser pour générer des données.

• Cliquez sur l’onglet **Preview** et cliquez sur le bouton **Preview** pour visualiser un échantillon des données qui seront générées.
Scénario : Génération aléatoire de fichiers de test

Le scénario suivant décrit un Job de deux composants, générant 50 lignes composées de la manière suivante : un ID aléatoire compris entre 1 et 3, un nom et un prénom pris de manière aléatoire dans une liste de chaînes de caractères, et le champ Date défini dans une période donnée.

- Cliquez et déposez un composant tRowGenerator et un composant tLogRow dans l'espace de modélisation.
- Cliquez-droit sur le composant tRowGenerator et sélectionnez le lien Row > Main. Faites glisser ce lien jusqu'au composant tLogRow.
- Double-cliquez sur le composant tRowGenerator pour ouvrir l'éditeur.
- Définissez les champs à générer.

<table>
<thead>
<tr>
<th>Schema</th>
<th>Functions</th>
<th>Preview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column</td>
<td>Key</td>
<td>Type</td>
</tr>
<tr>
<td>Random_ID</td>
<td>int</td>
<td>Integer</td>
</tr>
<tr>
<td>First_Name</td>
<td>String</td>
<td>String</td>
</tr>
<tr>
<td>Last_Name</td>
<td>String</td>
<td>String</td>
</tr>
<tr>
<td>Date</td>
<td>Date</td>
<td>Date</td>
</tr>
</tbody>
</table>

- La colonne ID (random_ID) est de type entier (integer), les noms et prénoms sont de type chaîne de caractères (String) et la Date est de type Date.
- Dans la colonne Function, sélectionnez la fonction adaptée dans la liste ou les trois points [...] pour la fonction personnalisée.
- Dans l'onglet Function parameters, définissez les valeurs à prendre de façon aléatoire.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>customize parameter</td>
<td>1,2,3</td>
<td></td>
</tr>
</tbody>
</table>

- Les colonnes First_Name et Last_Name sont à générer à l'aide d'une fonction getAsciiRandom String prédéfinie dans les routines système. Par défaut, la longueur définie est de 6 caractères, mais vous pouvez la changer.
- La colonne Date appelle également une fonction prédéfinie getRandomDate. Vous pouvez éditer les valeurs de paramètre dans l'onglet Function parameters.
- Dans le champ Number of Rows, paramétrez le nombre de lignes à 50.
- Cliquez sur OK pour valider les paramètres.
- Double-cliquez sur le composant tLogRow pour visualiser les paramètres dans l'onglet Basic settings. Retenez les paramètres par défaut.
- Appuyez sur F6 pour exécuter le Job.
Les 50 lignes sont générées selon les paramètres définis dans l'éditeur du composant tRowGenerator et sont affichées en sortie standard de la vue Run.
tRSSInput

Ce composant lit des flux RSS en utilisant des URL.

Le tRSSInput permet de suivre les billets postés sur certains sites en particulier afin d’en rassembler et organiser les informations rapidement et facilement pour votre utilisation.

**Propriétés du tRSSInput Standard**

Ces propriétés sont utilisées pour configurer le tRSSInput s’exécutant dans le framework de Jobs Standard.

Le composant tRSSInput Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Dans le contexte d’utilisation du tRSSInput, le schéma est composé de quatre colonnes : TITLE, DESCRIPTION, PUBDATE et Link. Les paramètres de titre sont en lecture seule, contrairement à leur type et leur longueur. |
| RSS URL | Saisissez l’URL du flux RSS à lire. |
| Read articles from | Lorsque cette case est cochée, le composant tRSSInput lit les articles du flux RSS à partir de la date renseignée en cliquant sur le bouton [...] situé à côté du champ date time. |
| Max number of articles | Lorsque cette case est cochée, le composant tRSSInput lit autant d’articles que le nombre spécifié dans le champ max amount. |
| Die on error | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et les lignes contenant les erreurs seront ignorées. |

**Global Variables**

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. |
| Global Variables |  |
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en tant que composant de début. Il requiert un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

### Scénario : Récupérer les billets fréquemment mis à jour sur un blog

Ce scénario Java à deux composants a pour but de récupérer les billets qui sont souvent mis à jour sur un blog, à partir d’un flux RSS local **Talend**, en utilisant le composant **tRSSInput**.

### Procédure

**Procédure**

1. A partir de la **Palette**, cliquez-déposez les composants suivants dans l’éditeur graphique : le **tRSSInput** et le **tLogRow**.
2. Cliquez-droit sur le premier pour le connecter au second à l’aide d’un lien de type **Row Main**.
3. Dans l’éditeur graphique, sélectionnez le **tRSSInput** et cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tRSSInput**.
4. Saisissez l'URL d'accès au flux RSS. Dans ce scénario, le **tRSSInput** est associé au flux RSS **Talend** : http://feeds.feedburner.com/Talend.

5. Cochez/décochez les autres cases selon vos besoins. Dans ce scénario, on veut afficher les informations de deux articles datés du 20 juillet 2008.

6. Dans l'espace graphique, sélectionnez le **tLogRow** puis cliquez sur la vue **Component** pour en définir la configuration de base. Pour plus d'informations sur les propriétés du composant **tLogRow**, consultez **Propriétés du tLogRow Standard** à la page 2105.

7. Enregistrez le Job puis appuyez sur **F6** pour l'exécuter.

```plaintext

#1. tLogRow_1

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>Welcoming Jean-Luc Solans</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Jean-Luc Solans joins Talend as VP of Strategy and Business</td>
</tr>
<tr>
<td>PUBLDATE</td>
<td>24 Jul 2008 19:40:11 GMT</td>
</tr>
<tr>
<td>LINK</td>
<td><a href="http://feeds.feedburner.com/~r/Talend/~3/344920575/">http://feeds.feedburner.com/~r/Talend/~3/344920575/</a></td>
</tr>
</tbody>
</table>

#2. tLogRow_1

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>Talend Open Profiler gets rave reviews</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>An interview and a product review of Talend Open Profiler</td>
</tr>
<tr>
<td>PUBLDATE</td>
<td>23 Jul 2008 21:12:53 GMT</td>
</tr>
<tr>
<td>LINK</td>
<td><a href="http://feeds.feedburner.com/~r/Talend/~3/343920856/">http://feeds.feedburner.com/~r/Talend/~3/343920856/</a></td>
</tr>
</tbody>
</table>

Job RSSInput ended at 15:17 07/08/2008. [exit code=0]

Le composant **tRSSInput** a accédé au flux RSS du site de **Talend** à votre place et en a organisé les informations pour vous.

Résultats

Deux billets sont affichés sur la console. Chacun d'eux a ses propres titre, description, date de publication et URL du flux RSS correspondante. Les blogs présentent le dernier billet en premier et vous pouvez faire défiler vers le bas pour lire les billets les plus anciens.
Ce composant crée et écrit des fichiers XML contenant des flux RSS ou Atom.

### Propriétés du tRSSOutput Standard

Ces propriétés sont utilisées pour configurer le tRSSOutput s’exécutant dans le framework de Jobs Standard.

Le composant tRSSOutput Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

#### Basic settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File Name</strong></td>
<td>Chemin d’accès et nom du fichier à traiter, et/ou variable à utiliser. For plus d’informations concernant l’utilisation et la définition de variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.</td>
</tr>
<tr>
<td><strong>Append</strong></td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
<tr>
<td><strong>Mode</strong></td>
<td>Sélectionnez RSS ou ATOM selon le flux que vous souhaitez générer.</td>
</tr>
</tbody>
</table>

**Channel** (en mode RSS)

- **Remarque** :
  - Les informations à renseigner dans cette zone concernent l’intégralité des données saisies, du site, etc. plutôt que des éléments particuliers.
- **Title** : Saisissez un titre significatif.
- **Description** : Entrez une description du contenu qui vous semble suffisamment détaillée.
- **Publication date** : Renseignez la date adéquate.
- **Link** : Saisissez l’URL appropriée.

**Feed** (en mode ATOM)

- **Title** : Saisissez un titre significatif.
- **Link** : Saisissez l’URL appropriée.
- **Id** : Saisissez l’URL valide correspondant au champ Link.
- **Update date** : Saisissez la date de mise à jour.
- **Author name** : Saisissez le nom de l’auteur.

**Optionnal Channel Elements**

Renseignez dans le tableau les informations relatives aux métadonnées du flux RSS. Cliquez sur le bouton [+] pour ajouter de nouvelles lignes.
**Element Name**: nom de la métadonnée.

**Element Value**: contenu de la métadonnée.

**Schema type et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Dans le contexte d’utilisation du tRSSinput, le schéma est composé de quatre colonnes : **TITLE**, **DESCRIPTION**, **PUBDATE** et **Link**. Les paramètres de titre sont en lecture seule, contrairement à leur type et leur longueur.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

**Global Variables**

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Utilisation

Règle d'utilisation  
Ce composant doit obligatoirement être relié à un composant de début ou un composant intermédiaire.

Scénario 1 : Créer un flux RSS et stocker les fichiers sur un serveur FTP

Ce scénario Java a pour but de :
• créer un flux RSS pour des fichiers que vous souhaitez partager avec d'autres personnes, et
• stocker l'ensemble des fichiers sur un serveur FTP.

Ce scénario permet d'écrire un fichier XML de flux RSS à partir d'une table MySQL contenant des informations concernant des livres. Il ajoute également des liens vers les fichiers stockés sur un serveur FTP au cas où les utilisateurs voudraient avoir accès aux fichiers dans leur intégralité.

Déposer et relier les composants

Procédure
1. A partir de la Palette, cliquez-déposez les composants suivants dans l'éditeur graphique : le tMysqlInput, le tRSSOutput et le tFTPPut.  
2. Cliquez-droit sur le tMysqlInput et connectez-le au tRSSOutput à l'aide d'un lien de type Row Main. 
3. Cliquez-droit sur le tMysqlInput et connectez-le au tFTPPut à l'aide d'un lien de type OnSubjobOk.

Définir la source des données d'entrée

Procédure
1. Dans l'espace graphique, sélectionnez le tMysqlInput.  
2. Cliquez sur la vue Component pour définir la configuration de base (Basic settings) du tMysqlInput.
3. Configurez le champ **Property type** en mode **Repository** puis cliquez sur le bouton [...] pour sélectionner la base de données adéquate dans la liste. Les détails de connexion ainsi que le schéma sont remplis automatiquement.

4. Dans le champ **Table Name**, saisissez le nom de votre table ou cliquez sur le bouton [...] et sélectionnez le nom de la table à partir de la liste. Dans ce scénario, la table Mysql de données saisies s'appelle "rss_talend" et le schéma comporte quatre colonnes, **TITLE**, **Description**, **PUBDATE**, et **LINK**.

5. Dans le champ **Query**, saisissez la requête de base de données en faisant particulièrement attention à ce que l’ordre des champs corresponde au schéma défini, ou cliquez sur **Guess Query**.

**Créer un flux RSS**

**Procédure**

1. Dans l’espace graphique, sélectionnez le composant **tRSSOutput**.
2. Cliquez sur la vue **Component** pour définir la configuration de base (**Basic settings**) du **tRSSOutput**.
3. Dans le champ **File name**, utilisez le nom de fichier et son chemin d'accès définis par défaut ou parcourrez vos dossiers pour définir votre propre fichier XML de sortie.

4. Sélectionnez le type d'encodage à partir de la liste **Encoding Type**.

5. Dans la zone **Mode**, sélectionnez **RSS**.

6. Dans le panneau **Channel**, saisissez un titre, une description, une date de publication et un lien définissant l'ensemble de vos données saisies.

7. Sélectionnez le type de schéma dans la liste **Schema** puis cliquez sur **Edit Schema** pour le modifier si nécessaire.

   **Remarque :**
   Vous pouvez également cliquer sur **Sync Column** pour récupérer le schéma généré par le composant précédent.

8. Enregistrez votre Job puis appuyez sur **F6** pour en exécuter la première partie.
Le composant tRSSOutput a créé en sortie un flux RSS au format XML pour les fichiers définis.

**Ecrire des fichiers entiers sur un serveur FTP**

**Pourquoi et quand exécuter cette tâche**

Pour stocker des fichiers complets sur un serveur FTP :

**Procédure**

1. Dans l'espace graphique, sélectionnez le FTPPut et cliquez sur la vue Component pour définir la configuration de base (Basic settings) du tFTPPut.
2. Saisissez le nom du serveur et le numéro de port dans les champs "Host" et "Port".
3. Renseignez vos détails de connexion dans les champs "Username" et "Password".
4. Parcourez vos dossier jusqu’au répertoire local ou tapez manuellement le chemin d’accès dans le champ "Local directory".
5. Entrez les détails du serveur de répertoire distant.
6. Sélectionnez le mode de transfert à partir de la liste "Transfer mode".
7. Dans le panneau "Files", cliquez sur le bouton "+" pour ajouter des lignes puis remplissez les masques de fichiers de tous les fichiers à copier sur le répertoire distant. Dans ce scénario, les fichiers à enregistrer sur le serveur FTP sont tous des fichiers texte.
8. Enregistrez votre Job puis appuyez sur F6 pour l’exécuter.

Résultats
Les fichiers définis sont copiés sur le serveur distant.

Scénario 2 : Créer un flux RSS contenant des métadonnées

Ce scénario Java à deux composants a pour but de créer un flux RSS contenant des métadonnées et de rediriger les informations obtenues dans un fichier XML de flux RSS en sortie.

Déposer et relier les composants

Procédure
A partir de la Palette, cliquez-déposez les composants tRSSInput et tRSSOutput dans l’éditeur graphique.

Cliquez-droit sur le tRSSInput et connectez-le au tRSSOutput à l’aide d’un lien de type "Row Main".
Configurer les composants

Procédure

1. Double-cliquez sur le composant tRSSInput pour ouvrir sa vue Basic settings et définir sa configuration de base.


3. Dans l’espace graphique, double-cliquez sur le composant pour ouvrir sa vue Component et définir sa configuration de base (Basic settings).

4. Dans le champ File name, utilisez le nom de fichier de sortie et son chemin d’accès définis par défaut ou parcourez vos dossiers pour définir votre propre fichier XML de sortie.

5. Sélectionnez le type d’encodage à partir de la liste Encoding Type.

6. Dans la zone Mode, sélectionnez RSS.

7. Dans le panneau Channel, saisissez un titre, une description, une date de publication et un lien définissant l’ensemble de vos données saisies.
Dans le tableau **Optional Channel Element**, définissez les métadonnées de votre flux RSS. Dans ce scénario, le flux comprend deux métadonnées : *copyright*, à laquelle est associée la valeur `tos`, et *language*, à laquelle est associée la valeur `en_us`.

9. Sélectionnez le type de schéma dans la liste **Schema** puis cliquez sur **Edit Schema** pour le modifier si nécessaire.

**Remarque :**

Vous pouvez également cliquer sur **Sync Column** pour récupérer le schéma généré par le composant précédent.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter.

**Résultats**

Les fichiers définis sont copiés dans le fichier XML de sortie et les métadonnées apparaissent sous le nœud `<channel>`, en dessous des informations de base du flux RSS.
Scénario 3 : Créer un fichier XML de flux ATOM

Ce scénario Java décrit un Job à deux composants, qui génère des données et les écrit dans un fichier XML de flux ATOM.

Déposer et relier les composants

Procédure
1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : un tFixedFlowInput du groupe Misc et un tRSSOutput du groupe Internet.
2. Cliquez-droit sur le tFixedFlowInput et connectez-le au tRSSOutput à l'aide d'un lien Row Main.
3. Une fenêtre pop-up s'affiche et vous demande si vous souhaitez passer le schéma du tRSSOutput au tFixedFlowInput, cliquez sur Yes (Oui).

Configurer les composants

Procédure
1. Dans l'espace de modélisation graphique, double-cliquez sur le tFixedFlowInput afin d'afficher sa vue Component et définir ses Basic settings.
2. Dans le champ Number of rows, laissez le paramètre par défaut, 1, pour ne générer qu'une ligne de données.
3. Dans la zone Mode, laissez sélectionnée l'option Use Single Table et remplissez le tableau Values. Notez que le champ Column du tableau Values est rempli par les colonnes du schéma défini dans le composant.
4. Dans le champ **Value** du tableau **Values**, saisissez les données que vous souhaitez envoyer au composant suivant.

5. Dans l'espace de modélisation, double-cliquez sur le composant **tRSSOutput** afin d'afficher sa vue **Component** et définir ses **Basic settings**.

6. Cliquez sur le bouton [...] à côté du champ **File Name** afin de définir le répertoire et le nom du fichier XML de sortie.

7. Dans la zone **Mode**, sélectionnez **ATOM** pour générer un fichier XML de flux ATOM.

   **Avertissement** :
   
   Comme le format du flux ATOM est strict, des informations par défaut sont requises afin de créer le fichier XML. Le schéma du **tRSSOutput** comprend donc des colonnes par défaut contenant ces informations. Ces colonnes par défaut sont grises pour indiquer qu'elles ne doivent pas être modifiées. Si vous choisissez de modifier le schéma du composant, le fichier XML ATOM créé ne sera pas valide.

8. Dans la zone **Feed**, saisissez un titre, un lien, un ID, la date de mise à jour, et le nom de l'auteur, afin de définir vos données comme un tout.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder votre Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de l'onglet **Run** pour l'exécuter.
Résultats

Le composant tRSSOutput crée un flux de sortie ATOM au format XML.
tRunJob

Ce composant gère des systèmes de Job complexes devant être exécuter l’un après l’autre.
Le composant tRunJob exécute le Job spécifié dans ses propriétés dans un contexte défini.

Propriétés du tRunJob Standard

Ces propriétés sont utilisées pour configurer le tRunJob s’exécutant dans le framework de Jobs Standard.
Le composant tRunJob Standard appartient aux familles System et Orchestration.
Le composant de ce framework est toujours disponible.

Avertissement :

Le composant tRunJob est supporté avec certaines limites, ce qui signifie que seuls les cas de support S4 (Minor) sont acceptés et qu’aucun patch n’est fourni. Si vous utilisez le tRunJob dans des Services de données et des Routes (avec le cTalendJob), le support est fourni uniquement sur la base des meilleurs efforts de l’entreprise. Dans la plupart des cas, des problèmes de chargements de classes peuvent survenir. Certains de ces problèmes peuvent être résolu, mais pas tous.

Comme le tRunJob n’est pas conçu pour fonctionner dans un déploiement de style Service/Route (ESB), le support habituel n’est pas fourni si vous décidez de l’utiliser, même si cela peut fonctionner dans de nombreux cas. Si vous avez utilisé le tRunJob par le passé, il est recommandé de modifier vos Jobs pour utiliser des Joblets à la place.

Pour des cas d’utilisation dans le domaine de l’intégration de données et non dans les domaines de l’ESB, ce composant est un composant très utile dont le support est fourni.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
| | • View schema : sélectionnez cette option afin de voir le schéma.
| | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
<p>| | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la |</p>
<table>
<thead>
<tr>
<th><strong>métadonnée du schéma dans la fenêtre [Repository Content].</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Copy Child Job Schema</strong></td>
</tr>
<tr>
<td><strong>Use dynamic job</strong></td>
</tr>
<tr>
<td><strong>Avertissement</strong> : L’option Use dynamic job n’est pas compatible avec le cache du JobServer. Dès lors, l’exécution peut échouer si vous exécutez dans Talend Administration Center un Job contenant un composant tRunJob avec cette option activée.</td>
</tr>
<tr>
<td><strong>Avertissement</strong> : Cette option est incompatible avec l’option Use or register a shared DB Connection des composants de base de données. Quand le tRunJob fonctionne de paire avec un composant de connexion à une base de données, activer les deux options fera échouer l’exécution du Job.</td>
</tr>
<tr>
<td><strong>Avertissement</strong> : Cette option n’est pas supportée dans les Routes ou les Services de données ESB.</td>
</tr>
<tr>
<td><strong>Context job</strong></td>
</tr>
<tr>
<td><strong>Job</strong></td>
</tr>
<tr>
<td><strong>Version</strong></td>
</tr>
<tr>
<td><strong>Context</strong></td>
</tr>
<tr>
<td><strong>Use an independent process to run subjob</strong></td>
</tr>
</tbody>
</table>
### Avertissement : L’option *Use dynamic job* n’est pas compatible avec le cache du JobServer. Dès lors, l’exécution peut échouer si vous exécutez dans *Talend Administration Center* un Job contenant un composant **tRunJob** avec cette option activée.

### Avertissement : Cette option n’est pas compatible avec l’option *Use or register a shared DB Connection* des composants de connexion à une base de données. Lorsque le **tRunJob** fonctionne avec un composant de connexion à une base de données, activer les deux options peut faire échouer votre Job.

<table>
<thead>
<tr>
<th><strong>Die on child error</strong></th>
<th>Décochez cette case pour exécuter le Job parent même s’il y a une erreur lors de l’exécution du Job fils.</th>
</tr>
</thead>
</table>
| **Transmit whole context** | Cochez cette case pour obtenir les contextes du Job parent. Décochez-la pour obtenir les contextes du Job fils.  
Si cette case est cochée lorsque les Jobs père et fils ont les mêmes variables de contexte définies :
- les valeurs des variables pour le Job père sont utilisées durant l’exécution du Job fils si aucune valeur cohérente n’est définie dans la table *Context Param*.  
- sinon, les valeurs définies dans la table *Context Param* sont utilisées durant l’exécution du Job fils. |
| **Context Param** | Vous pouvez modifier la valeur du contexte sélectionné. Cliquez sur le bouton [+] pour ajouter les paramètres définis dans l’onglet *Context contexte du Job fils*. Pour plus d’informations sur les paramètres de contexte, consultez le *Guide utilisateur du Studio Talend*.  
Les valeurs définies ici sont utilisées durant l’exécution du Job fils, même si la case *Transmit whole context* est cochée. |

### Advanced settings

| **Propagate the child result to the output schema** | Cochez cette case afin de propager les données de sortie stockées dans la mémoire tampon via le composant **tBufferOutput** du Job fils vers le composant de sortie du Job père.  
Cette case est décochée par défaut. Elle ne s’affiche pas lorsque vous cochez la case *Use dynamic job* ou *Use an independent process to run subjob*. |
| **Print Parameters** | Cochez cette case pour afficher les paramètres internes et externes dans la *Console*. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |
### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td>CHILD_RETURN_CODE</td>
<td>code de retour d’un Job fils. Cette variable est une variable After et retourne un nombre entier.</td>
</tr>
<tr>
<td>CHILD_EXCEPTION_STACKTRACE</td>
<td>trace de la pile d’un Job fils. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant peut être utilisé en Job standalone ou peut aider à clarifier certains Jobs complexes en évitant d’avoir un nombre important de sous-jobs dans le même Job.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connections</th>
<th>DESCRIPTION</th>
</tr>
</thead>
</table>

Pour plus d’informations concernant les liens, consultez la section relative aux types de connexions, dans le Guide utilisateur du Studio Talend.

### Appeler un Job et passer le paramètre nécessaire au Job appelé

Ce scénario décrit un Job à deux composants, nommé ParentJob, appelant un autre Job nommé ChildJob afin d’afficher le contenu des fichiers spécifiés dans le Job ParentJob, dans la console de la vue Run.
Configurer le Job fils

Procédure

2. Reliez le tFileInputDelimited au tLogRow à l'aide d'un lien Row > Main.
3. Double-cliquez sur le tFileInputDelimited pour ouvrir sa vue Basic settings.
4. Cliquez dans le champ File Name, appuyez sur F5 pour ouvrir la boîte de dialogue New Context Parameter et configurer la variable de contexte.
5. Dans le champ **Name**, saisissez un nom pour cette variable de contexte, **FilePath**, dans cet exemple.

6. Dans le champ **Default value**, saisissez le chemin d'accès complet au fichier d'entrée par défaut.

7. Cliquez sur **Finish** pour valider la configuration du paramètre de contexte et renseigner le champ **File Name** avec la variable de contexte.

   Vous pouvez également créer ou modifier un paramètre de contexte dans l’onglet **Context** sous l’espace de modélisation graphique. Pour plus d’informations, consultez le Guide utilisateur du Studio Talend.

8. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir la fenêtre **Schema**, dans laquelle vous pourrez configurer le schéma manuellement.

9. Dans la boîte de dialogue, cliquez sur le bouton [+] pour ajouter des colonnes et nommez-les en respectant la structure du fichier d’entrée.

   Dans cet exemple, ce composant lit les fichiers définis dans le Job père. Ces fichiers contiennent jusqu’à cinq colonnes. Ajoutez cinq colonnes de type String et nommez-les **Column1**, **Column2**, **Column3**, **Column4** et **Column5** respectivement. Cliquez sur OK pour valider la configuration du schéma et fermer la fenêtre **Schema**.

10. Double-cliquez sur le composant **tLogRow** et, dans sa vue **Basic settings**, sélectionnez l’option **Table** pour voir le contenu s’afficher sous forme de tableau.

### Configurer le Job père

**Procédure**

1. Créez un nouveau Job père **ParentJob** et ajoutez-y un **tFileList** et un **tRunJob**.

2. Reliez le **tFileList** au **tRunJob** à l’aide d’un lien **Row>Iterate**.

3. Double-cliquez sur le **tFileList** pour ouvrir sa vue **Basic settings**.
4. Dans le champ **Directory**, spécifiez le chemin d'accès au répertoire contenant les fichiers à traiter, ou cliquez sur le bouton [...] à côté du champ pour parcourir votre répertoire.

Dans cet exemple, le répertoire est D:/tRunJob_Input_Files, contenant trois fichiers délimités ayant chacun cinq colonnes maximum.

5. Dans la liste **FileList Type**, sélectionnez Files.

6. Cochez la case **Use Glob Expressions as Filemask**, puis cliquez sur le bouton [+ ] pour ajouter une ligne dans la table Files et définir un filtre sur les fichiers. Dans cet exemple, saisissez "*.csv" afin de récupérer tous les fichiers délimités.

7. Double-cliquez sur le composant tRunJob pour ouvrir sa vue **Basic settings**.

8. Cliquez sur le bouton [...] à côté du champ **Job** et, dans la boîte de dialogue, sélectionnez le Job fils à exécuter, puis cliquez sur OK pour fermer la boîte de dialogue. Le nom du Job sélectionné s'affiche dans le champ **Job**.
9. Dans la table **Context Param**, cliquez sur le bouton [+ ] pour ajouter une ligne et définir le paramètre de contexte. Le seul paramètre de contexte défini dans le Job fils, nommé **FilePath**, s’affiche dans la cellule **Parameters**.

10. Cliquez dans la cellule **Values**, appuyez sur les touches **Ctrl+Espace** de votre clavier, afin d’accéder à la liste des variables de contexte et sélectionnez **tFileList_1.CURRENT_FILEPATH**.

La variable de contexte `((String)globalMap.get("tFileList_1_CURRENT_FILEPATH"))` correspondante s’affiche dans la cellule **Values**.

Pour plus d’informations concernant les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

**Exécuter le Job père**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder vos Jobs.

2. Appuyez sur **F6** pour exécuter le Job père.

Le Job père appelle le Job fils, qui lit les fichiers définis dans le Job père. Le contenu des fichiers est affiché dans la console de la vue **Run**.
Exécuter dynamiquement une liste de Jobs fils

Ce scénario décrit un Job appelant deux Jobs fils de manière dynamique. Lorsqu'il est appelé depuis le Job père, chaque Job fils affiche un message dans la console.

Configurer les Jobs fils

Procédure

1. Créez un nouveau Job nommé ChildJob1 et ajoutez-y un composant tFixedFlowInput et un tLogRow.

![Diagramme des Jobs fils](image)

2. Reliez le tFixedFlowInput au tLogRow à l'aide d'un lien Row > Main.

4. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la fenêtre qui s'affiche, définissez le schéma des données d'entrée en ajoutant une colonne Message de type String. Cela fait, cliquez sur OK pour fermer la fenêtre et cliquez sur Yes afin de propager le schéma au composant suivant.
5. Dans la zone Mode, sélectionnez Use Inline Content(delimited file) et saisissez dans le champ Content le message à afficher dans la console, Hello World! dans cet exemple.
6. Double-cliquez sur le composant tLogRow et, dans sa vue Basic settings, sélectionnez le mode Table pour afficher les résultats d'exécution sous forme de tableau.

Configurer le Job père

Procédure

1. Créez un nouveau Job nommé ParentJob et ajoutez-y un tFixedFlowInput, un tFlowToIterate et un tRunJob.
2. Reliez le composant `tFixedFlowInput` au `tFlowToIterate` à l’aide d’un lien `Row > Main` et le `tFlowToIterate` au `tRunJob` à l’aide d’un lien `Row > Iterate`.

3. Double-cliquez sur le `tFixedFlowInput` pour ouvrir sa vue `Basic settings`.

4. Cliquez sur le bouton ` [...] ` à côté du champ `Edit schema` et, dans la fenêtre qui s’ouvre, définissez le schéma des données d’entrée en ajoutant une colonne `JobName` de type `String`. Cela fait, cliquez sur `OK` pour fermer la boîte de dialogue.

5. Dans la zone `Mode`, sélectionnez l’option `Use Inline Content(delimited file)` et spécifiez les noms des Jobs fils à appeler du Job père dans le champ `Content`.

6. Double-cliquez sur le composant `tRunJob` pour ouvrir sa vue `Basic settings`.

7. Cochez la case `Use dynamic job` et, dans le champ `Context job` qui s’affiche, appuyez sur les touches `Ctrl+Espace`. Dans la liste des variable, sélectionnez la variable globale itérative créée...
par le tFlowToIterate, tFlowToIterate_1.JobName dans cet exemple. Le champ Context job est ensuite renseigné avec ((String)globalMap.get("row1.JobName"))). À chaque itération, cette variable sera résolue comme le nom du Job à appeler.


Exécuter le Job père pour exécuter dynamiquement les Jobs fils

Procédure

1. Sauvegardez vos Jobs fils et père.
2. Appuyez sur F6 ou cliquez sur le bouton Run de la vue Run pour exécuter le Job.

Comme affiché ci-dessus, les Jobs fils sont appelés l’un après l’autre et les messages spécifiés sont affichés dans la console.
**Propager les données de sortie en mémoire tampon du Job fils au Job père**

Dans ce scénario, un Job a trois composants appelle un Job fils à deux composants et affiche dans la console les données de sortie mises en mémoire tampon du Job fils et non les données du flux d’entrée du Job père.

**Configurer le Job fils**

**Pourquoi et quand exécuter cette tâche**

![Diagramme de configuration du Job fils](image)

**Procédure**

1. Créez un Job nommé `child` et ajoutez-y deux composants en saisissant leur nom dans l’espace de modélisation graphique, ou en les déposant depuis la Palette dans l’espace de modélisation graphique :
   - un `tFixedFlowInput`, afin de générer un message,
   - un `tBufferOutput`, afin de stocker le message généré dans la mémoire tampon.
2. Reliez le `tFixedFlowInput` au `tBufferOutput`, à l’aide d’un lien `Row > Main`.

![Vues et options du composant tFixedFlowInput](image)

Cela fait, cliquez sur **OK** afin de valider les modifications, puis cliquez sur **Yes** dans la boîte de dialogue **[Propagate]**, afin de propager le schéma au composant suivant.

5. Dans la zone **Mode**, sélectionnez l’option **Use Single Table** et définissez la valeur correspondante pour la colonne **message** dans la table **Values**. Dans cet exemple, la valeur est "message from the child job".

### Configurer le Job père

#### Pourquoi et quand exécuter cette tâche

![Diagramme de flux]

1. Créez un Job et ajoutez-y trois composants, en saisissant leur nom dans l’espace de modélisation graphique en glissant-déposant depuis la **Palette** dans l’espace de modélisation graphique :
   - un **tFixedFlowInput**, pour générer un message,
   - un **tRunJob**, pour appeler le Job nommé **child**,
   - un **tLogRow**, pour afficher les résultats d’exécution dans la console.
2. Reliez le **tFixedFlowInput** au **tRunJob** et le **tRunJob** au **tLogRow** à l’aide de liens **Row > Main**.
3. Double-cliquez sur le composant **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

Cela fait, cliquez sur OK pour valider les modifications.

5. Dans la zone Mode, sélectionnez l’option Use Single Table et définissez la valeur correspondante pour la colonne message dans la table Values. Dans cet exemple, la valeur est “message from the parent job”.

6. Cliquez sur le tRunJob et cliquez sur l’onglet Component pour ouvrir sa vue Basic settings.
7. Cliquez sur le bouton **Sync columns**, puis cliquez sur **Yes** dans la boîte de dialogue **[Propagate]** qui s'ouvre afin de récupérer le schéma du composant précédent.

8. Cliquez sur le bouton 

   ![Sync columns button](image)

   à côté du champ **Job** pour ouvrir la boîte de dialogue **[Repository Content]**.

   ![Repository Content](image)

   Dans la boîte de dialogue **[Repository Content]**, sélectionnez le Job nommé *child* puis cliquez sur **OK** pour fermer la boîte de dialogue.

9. Dans la vue **Advanced settings** du **tRunJob**, cochez la case **Propagate the child result to the output schema**. Cette case cochée, les données de sortie en mémoire tampon du Job fils seront propagées au composant de sortie.

   ![Advanced settings](image)

### Exécuter le Job père

**Procédure**

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder le Job.

2. Appuyez sur **F6** ou cliquez sur le bouton **Run** de la vue **Run** pour exécuter le Job.

   ![Execution](image)

   _Starting job parent at 10:13 16/04/2014._

   ![Job output](image)

   Le Job fils est appelé et le message défini dans le Job fils s'affiche dans la console et non le message défini dans le Job père.
**tS3BucketCreate**

Ce composant crée un bucket dans Amazon S3.

"Bucket" est un terme utilisé par AWS pour les dossiers du plus haut niveau sur S3, contenant des sous-dossiers et stockant toutes vos données (objets).

**Propriétés du tS3BucketCreate Standard**

Ces propriétés sont utilisées pour configurer le tS3BucketCreate s’exécutant dans le framework de Jobs Standard.

Le composant tS3BucketCreate Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Use existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Key</td>
<td>L’ID de la clé d’accès (Access Key) identifie uniquement un compte AWS. Pour plus d’informations concernant la clé d’accès et sa sécurité, consultez le site Obtention de vos clés d’accès AWS.</td>
</tr>
<tr>
<td>Secret Key</td>
<td>La Secret Access Key, combinée à la clé d’accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour obtenir les informations d’authentification de sécurité AWS depuis la métadonnée de l’instance Amazon EC2. Pour utiliser cette option, l’instance Amazon EC2 doit être démarrée et votre Job doit être en cours d’exécution sur Amazon EC2. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
</tbody>
</table>
| Assume role             | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  
  - **Role ARN** : nom Amazon Resource Name (ARN) du rôle.  
  - **Role session name** : identifiant de la session du rôle.  
  - **Session duration (minutes)** : durée (en minutes) pour laquelle est active la session du rôle. |
Pour plus d'informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).

### Region
Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles ("us-east-1" par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Régions et points de terminaison AWS.

### Bucket
Nom du bucket, le dossier de plus haut niveau, à créer.

### Die on error
Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

### Advanced settings

| Config client | Cochez cette case pour configurer des paramètres client.  
  **Client parameter** : sélectionnez les paramètres client dans la liste.  
  **Value** : saisissez la valeur des paramètres.  
  Cette option n'est pas disponible lorsque la case Use existing connection est cochée. |
|---------------|---------------------------------------------------------------------------------------------------------------|
| STS Endpoint  | Cochez cette case et, dans le champ qui s'affiche, spécifiez l'endpoint du service AWS Security Token Service duquel les informations d'authentification sont récupérées.  
  Cette case est disponible uniquement lorsque la case Assume role est cochée. |
| tStatCatcher Statistics | Cochez cette case afin de collecter les données de log au niveau des composants. |

### Global Variables

<table>
<thead>
<tr>
<th>ERROR_MESSAGE</th>
<th>Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case Die on error est cochée.</th>
</tr>
</thead>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant peut être utilisé en standalone ou avec d'autres composants S3, par exemple le tS3BucketExist.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant</td>
</tr>
</tbody>
</table>
la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Pour un scénario associé au tS3BucketCreate, consultez Scénario : Vérifier l’absence d’un bucket, le créer et lister tous les buckets S3 à la page 3521.
**tS3BucketDelete**

Ce composant supprime un bucket vide d’Amazon S3.

“Bucket” est un terme utilisé par AWS pour les dossiers du plus haut niveau sur S3, contenant des sous-dossiers et stockant toutes vos données (objets).

**Propriétés du tS3BucketDelete Standard**

Ces propriétés sont utilisées pour configurer le tS3BucketDelete s’exécutant dans le framework de Jobs Standard.

Le composant tS3BucketDelete Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Access Key</strong></td>
<td>L’ID de la clé d’accès (Access Key) identifie uniquement un compte AWS. Pour plus d’informations concernant la clé d’accès et sa sécurité, consultez le site Obtention de vos clés d’accès AWS.</td>
</tr>
<tr>
<td><strong>Secret Key</strong></td>
<td>La Secret Access Key, combinée à la clé d’accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Inherit credentials from AWS role</strong></td>
<td>Cochez cette case pour obtenir les informations d’authentification de sécurité AWS depuis la métadonnée de l’instance Amazon EC2. Pour utiliser cette option, l’instance Amazon EC2 doit être démarrée et votre Job doit être en cours d’exécution sur Amazon EC2. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
<tr>
<td><strong>Region</strong></td>
<td>Spécifiez la zone géographique AWS en sélectionnant le nom d’une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (“us-east-1” par exemple) dans la liste. Pour plus d’informations sur les zones géographiques AWS, consultez Régions et points de terminaison AWS.</td>
</tr>
<tr>
<td><strong>Bucket</strong></td>
<td>Nom du bucket, le dossier de plus haut niveau, à supprimer.</td>
</tr>
</tbody>
</table>
### Die on error

Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

### Advanced settings

#### Config client

Cochez cette case pour configurer des paramètres client.

- **Client parameter** : sélectionnez les paramètres client dans la liste.
- **Value** : saisissez la valeur des paramètres.

Cette option n’est pas disponible lorsque la case **Use existing connection** est cochée.

#### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau des composants.

### Variables globales

#### ERROR_MESSAGE

Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case **Die on error** est cochée.

### Utilisation

#### Règle d’utilisation

Ce composant peut être utilisé en standalone ou avec d’autres composants S3, par exemple le **tS3BucketList**.

#### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les
paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Scénario associé

Pour un scénario associé au **tS3BucketDelete**, consultez *Scénario : Vérifier l’absence d’un bucket, le créer et lister tous les buckets S3* à la page 3521.
tS3BucketExist

Ce composant vérifie que le bucket spécifié existe dans Amazon S3.

"Bucket" est un terme utilisé par AWS pour les dossiers du plus haut niveau sur S3, contenant des sous-dossiers et stockant toutes vos données (objets).

**Propriétés du tS3BucketExist Standard**

Ces propriétés sont utilisées pour configurer le tS3BucketExist s'exécutant dans le framework de Jobs Standard.

Le composant tS3BucketExist Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Use existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Key</td>
<td>L’ID de la clé d’accès (Access Key) identifie uniquement un compte AWS. Pour plus d’informations concernant la clé d’accès et sa sécurité, consultez le site Obtention de vos clés d’accès AWS.</td>
</tr>
<tr>
<td>Secret Key</td>
<td>La Secret Access Key, combinée à la clé d’accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour obtenir les informations d’authentification de sécurité AWS depuis la métdonnée de l’instance Amazon EC2. Pour utiliser cette option, l’instance Amazon EC2 doit être démarrée et votre Job doit être en cours d’exécution sur Amazon EC2. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2 .</td>
</tr>
<tr>
<td>Assume role</td>
<td>Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Role ARN</strong> : nom Amazon Resource Name (ARN) du rôle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Role session name</strong> : identifiant de la session du rôle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Session duration (minutes)</strong> : durée (en minutes) pour laquelle est active la session du rôle.</td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).

### Region
Spécifiez la zone géographique AWS en sélectionnant le nom d’une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles ("us-east-1" par exemple) dans la liste. Pour plus d’informations sur les zones géographiques AWS, consultez Régions et points de terminaison AWS.

### Bucket
Nom du bucket, le dossier de plus haut niveau, sur le serveur S3.

### Die on error
Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

#### Advanced settings

| Config client | Cochez cette case pour configurer des paramètres client.  
| Client parameter : sélectionnez les paramètres client dans la liste.  
| Value : saisissez la valeur des paramètres.  
| Cette option n’est pas disponible lorsque la case Use existing connection est cochée. |
| STS Endpoint | Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.  
| Cette case est disponible uniquement lorsque la case Assume role est cochée. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau des composants. |

#### Variables globales

| BUCKET_EXIST | Indique si le bucket spécifié existe. Cette variable est une variable Flow et retourne un booléen. |
| BUCKET_NAME | Nom d’un bucket spécifié. Cette variable est une variable After et retourne une chaîne de caractères. |
| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case Die on error est cochée. |
# Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec d’autres composants S3, comme le tS3BucketCreate.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

## Scénario : Vérifier l’absence d’un bucket, le créer et lister tous les buckets S3

Dans ce scénario, le tS3BucketExist est utilisé pour vérifier l’absence d’un bucket. Le composant tS3BucketCreate est utilisé pour créer ce bucket après confirmation de l’absence du bucket. Le tS3BucketList permet de lister tous les buckets dans Amazon S3.

### Construire le Job

**Procédure**

1. Déposez un tS3Connection, un tS3BucketExist, un tS3BucketCreate, un tS3BucketList, un tIterateToFlow et un tLogRow dans l’espace de modélisation graphique.
2. Reliez le tS3Connection au tS3BucketExist à l’aide d’un lien OnSubjobOk.
3. Reliez le tS3BucketExist au tS3BucketCreate à l’aide d’un lien Run if.
4. Reliez le tS3BucketCreate au tS3BucketList à l’aide d’un lien OnSubjobOk.
5. Reliez le tS3BucketList au tIterateToFlow à l’aide d’un lien Row > Iterate.
6. Reliez le tIterateToFlow au tLogRow à l’aide d’un lien Row > Main.
Configurer les composants

Procédure

1. Double-cliquez sur le **tS3Connection** pour ouvrir sa vue **Basic settings**.

2. Dans les champs **Access Key** et **Secret Key**, saisissez vos informations d’authentification.

3. Double-cliquez sur le composant **tS3BucketExist** pour ouvrir sa vue **Basic settings**.

4. Cochez la case **Use existing connection** afin de réutiliser la connexion définie dans le composant précédent.

5. Dans le champ **Bucket**, saisissez le nom du bucket afin de vérifier son existence.

6. Double-cliquez sur le lien **If** afin de configurer sa condition.
7. Dans le champ **Condition**, saisissez l’expression :

```
!((Boolean)globalMap.get("tS3BucketExist_1_BUCKET_EXIST"))
```

Ainsi, le reste du Job est exécuté si le bucket spécifié n’existe pas.

8. Double-cliquez sur le **tS3BucketCreate** pour ouvrir sa vue **Basic settings**.

Cochez la case **Use existing connection** afin de réutiliser la connexion.
Dans le champ **Bucket**, saisissez le nom du bucket à créer.

9. Double-cliquez sur le **tS3BucketList** pour ouvrir sa vue **Basic settings**.

Cochez la case **Use existing connection** pour réutiliser la connexion.

10. Double-cliquez sur le **tIterateToFlow** pour ouvrir sa vue **Basic settings**.

11. Cliquez sur le bouton **Edit schema** pour ouvrir l’éditeur de schéma.
Cliquez sur le bouton [+] pour ajouter une colonne. Nommez-la bucket_list et définissez son type à String.

Cliquez sur OK pour valider la configuration et fermer l’éditeur.

12. Dans la zone Mapping, appuyez sur les touches Ctrl+Espace dans le champ Value afin de sélectionner la variable tS3BucketList_1_CURRENT_BUCKET_NAME.


 Sélectionnez l’option Table (print values in cells of a table) pour un affichage optimal des résultats.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl + S afin de sauvegarder le Job.
2. Appuyez sur F6 pour exécuter le Job.

```
[statistics] connecting to socket on port 3661
[statistics] connected

<table>
<thead>
<tr>
<th>tLogRow_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>bucket_list</td>
</tr>
<tr>
<td>movies_andy</td>
</tr>
<tr>
<td>songs_andy</td>
</tr>
</tbody>
</table>

[statistics] disconnected
```

Comme affiché ci-dessus, le bucket est créé et tous les buckets sont listés.

3. Allez dans la console Web S3 :
Vous pouvez constater que le bucket a bien été créé sur le serveur S3.

4. Rafraîchissez le client S3 Browser :

La capture d’écran ci-dessus vous montre que l’action S3 Create s’est déroulée avec succès.
tS3BucketList

Ce composant liste tous les buckets dans Amazon S3.

"Bucket" est un terme utilisé par AWS pour les dossiers du plus haut niveau sur S3, contenant des sous-dossiers et stockant toutes vos données (objets).

**Propriétés du tS3BucketList Standard**

Ces propriétés sont utilisées pour configurer le tS3BucketList s’exécutant dans le framework de Jobs Standard.

Le composant tS3BucketList Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Proposition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Access Key</td>
<td>L’ID de la clé d’accès (Access Key) identifie uniquement un compte AWS. Pour plus d’informations concernant la clé d’accès et sa sécurité, consultez le site Obtention de vos clés d’accès AWS.</td>
</tr>
</tbody>
</table>
| Secret Key                              | La Secret Access Key, combinée à la clé d’accès, constitue votre accès sécurisé à Amazon S3.  
|                                        | Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres. |
| Inherit credentials from AWS role       | Cochez cette case pour obtenir les informations d’autheutification de sécurité AWS depuis la métadonnée de l’instance Amazon EC2. Pour utiliser cette option, l’instance Amazon EC2 doit être démarrée et votre Job doit être en cours d’exécution sur Amazon EC2. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2. |
| Assume role                             | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  
|                                        | • Role ARN : nom Amazon Resource Name (ARN) du rôle.  
|                                        | • Role session name : identifiant de la session du rôle.  
|                                        | • Session duration (minutes) : durée (en minutes) pour laquelle est active la session du rôle. |
Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).

Region

Spécifiez la zone géographique AWS en sélectionnant le nom d’une zone géographique dans la liste ou en saisisant le nom de la zone entre guillemets doubles ("us-east-1" par exemple) dans la liste. Pour plus d’informations sur les zones géographiques AWS, consultez Régions et points de terminaison AWS.

Advanced settings

Config client

Cochez cette case pour configurer des paramètres client.

Client parameter : sélectionnez les paramètres client dans la liste.

Value : saisissez la valeur des paramètres.

Cette option n’est pas disponible lorsque la case Use existing connection est cochée.

STS Endpoint

Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.

Cette case est disponible uniquement lorsque la case Assume role est cochée.

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau des composants.

Variables globales

CURRENT_BUCKET_NAME

Nom du bucket courant. Cette variable est une variable Flow et retourne une chaîne de caractères.

NB_BUCKET

Nombre de buckets. Cette variable est une variable After et retourne un entier.

ERROR_MESSAGE

Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

Utilisation

Règle d’utilisation

Ce composant peut être utilisé en standalone ou avec d’autres composants S3, comme le tS3BucketDelete.

Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez...
dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Pour un scénario associé au tS3BucketList, consultez Scénario : Vérifier l’absence d’un bucket, le créer et lister tous les buckets S3 à la page 3521.
**tS3Close**

Ce composant ferme une connexion à Amazon S3 et libère ainsi les ressources réseau.

**Propriétés du tS3Close Standard**

Ces propriétés sont utilisées pour configurer le tS3Close s’exécutant dans le framework de Jobs Standard.
Le composant tS3Close Standard appartient à la famille Cloud.
Le composant de ce framework est toujours disponible.

ts3close amazon_s3

**Basic settings**

<table>
<thead>
<tr>
<th>Component List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez dans la liste le composant de connexion.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour collecter les données de log au niveau des composants.</td>
</tr>
</tbody>
</table>

**Variables globales**

<table>
<thead>
<tr>
<th>ERROR_MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>En tant que composant de sortie, ce composant doit être utilisé avec d’autres composants S3, par exemple le tS3Connection.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.
Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

## Scénario associé

Pour un scénario associé au **tS3Close**, consultez **Scénario : Lister les fichiers d’un bucket ayant le même préfixe** à la page 3550.
**tS3Connection**

Ce composant tS3Connection permet de vous connecter à Amazon S3 afin de stocker et de récupérer des données.

### Propriétés du tS3Connection Standard

Ces propriétés sont utilisées pour configurer le tS3Connection s'exécutant dans le framework de Jobs Standard.

Le composant tS3Connection Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

#### Basic settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Key</td>
<td>L’ID de la clé d’accès (Access Key) identifie uniquement un compte AWS. Pour plus d’informations concernant la clé d’accès et sa sécurité, consultez le site Obtention de vos clés d’accès AWS.</td>
</tr>
<tr>
<td>Secret Key</td>
<td>La Secret Access Key, combinée à la clé d’accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour obtenir les informations d’authentification de sécurité AWS depuis la métadonnée de l’instance Amazon EC2. Pour utiliser cette option, l’instance Amazon EC2 doit être démarrée et votre Job doit être en cours d’exécution sur Amazon EC2. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
<tr>
<td>Assume role</td>
<td>Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Role ARN</strong> : nom Amazon Resource Name (ARN) du rôle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Role session name</strong> : identifiant de la session du rôle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Session duration (minutes)</strong> : durée (en minutes) pour laquelle est active la session du rôle.</td>
</tr>
<tr>
<td>Region</td>
<td>Spécifiez la zone géographique AWS en sélectionnant le nom d’une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (&quot;us-east-1&quot; par exemple) dans la liste. Pour</td>
</tr>
</tbody>
</table>
plus d’informations sur les zones géographiques AWS, consultez [Régions et points de terminaison AWS](#).

### Encrypt

Cochez cette case et dans la liste déroulante Key type affichée, sélectionnez l’une des trois options suivantes pour chiffrer les données côté client avant de les envoyer à Amazon S3. Pour plus d’informations, consultez [Protection des données via le chiffrement côté client](#).

- **KMS-managed customer master key** : utilisez une clé maître client (CMK) gérée par KMS pour le chiffrement des données côté client. Dans le champ Key, spécifiez l’identifiant de la clé maître client AWS gérée par KMS (CMK ID).
- **Symmetric Master Key** : utilisez une clé maître symétrique (clé secrète AES 256 bits) pour le chiffrement des données côté client.
  - **Algorithm** : sélectionnez l’algorithme associé à la clé dans la liste. Par défaut, il n’y a qu’un seul algorithme nommé AES.
  - **Encoding** : sélectionnez le type de chiffrement associé à la clé dans la liste, Base64 ou X509.
  - **Key or Key file** : spécifiez la clé ou le chemin du fichier contenant la clé.
- **Asymmetric Master Key** : utilisez une clé maître asymétrique (une paire de clés RSA 1024 bits) pour le chiffrement des données côté client.
  - **Algorithm** : sélectionnez l’algorithme associé à la clé dans la liste. Par défaut, il n’y a qu’un seul algorithme nommé RSA.
  - **Public key file** : spécifiez le chemin du fichier contenant la clé publique.
  - **Private key file** : spécifiez le chemin du fichier contenant la clé privée.

### Advanced settings

<table>
<thead>
<tr>
<th>Use a custom region endpoint</th>
<th>Cochez cette case pour utiliser un endpoint personnalisé et, dans le champ qui s’affiche, spécifiez l’URL de l’endpoint personnalisé à utiliser.</th>
</tr>
</thead>
</table>
| Config client                 | Cochez cette case pour configurer des paramètres client.  
**Client parameter** : sélectionnez les paramètres client dans la liste.  
**Value** : saisissez la valeur des paramètres. |
| STS Endpoint                  | Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.  
Cette case est disponible uniquement lorsque la case **Assume role** est cochée. |
tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau des composants.

Variables globales

ERROR_MESSAGE

Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

Utilisation

Règle d’utilisation

Ce composant est un composant de début, il doit être utilisé avec d’autres composants S3.

Dynamic settings

Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénario associé

Pour un scénario associé contenant un composant tS3Connection, consultez Scénario : Échange de fichiers avec Amazon S3 à la page 3558.
Ce composant copie un objet Amazon S3 d'un bucket source à un bucket cible.

**Propriétés du tS3Copy Standard**

Ces propriétés sont utilisées pour configurer le tS3Copy s'exécutant dans le framework de Jobs Standard.

Le composant tS3Copy Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access Key</strong></td>
<td>Spécifiez l'ID de la clé d'accès (Access Key) identifie uniquement un compte AWS. Pour plus d'informations concernant la clé d'accès et sa sécurité, consultez le site <a href="https://aws.amazon.com">Obtention de vos clés d'accès AWS</a>.</td>
</tr>
<tr>
<td><strong>Secret Key</strong></td>
<td>Spécifiez la Secret Access Key, qui, combinée à la clé d'accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ <strong>Secret key</strong>, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Inherit credentials from AWS role</strong></td>
<td>Cochez cette case pour obtenir les informations d'authentification de sécurité AWS depuis la métadonnée de l'instance Amazon EC2. Pour utiliser cette option, l'instance Amazon EC2 doit être démarrée et votre Job doit être en cours d'exécution sur Amazon EC2. Pour plus d'informations, consultez <a href="https://docs.aws.amazon.com/IAM/latest/userguide/idp-instance-principal.html">Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2</a>.</td>
</tr>
</tbody>
</table>
| **Assume role**            | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  
  - **Role ARN** : nom Amazon Resource Name (ARN) du rôle.  
  - **Role session name** : identifiant de la session du rôle.  
  - **Session duration (minutes)** : durée (en minutes) pour laquelle est active la session du rôle.  
  Pour plus d'informations concernant les rôles et AssumeRole, consultez [AssumeRole](https://docs.aws.amazon.com/IAM/latest/userguide/idp-instance-principal.html) (en anglais). |
| Region | Spécifiez la zone géographique AWS en sélectionnant le nom d’une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles ("us-east-1" par exemple) dans la liste. Pour plus d’informations sur les zones géographiques AWS, consultez Régions et points de terminaison AWS.

| Source Bucket | Spécifiez le nom du bucket source contenant l’objet à copier.

| Source Key | Spécifiez la clé de l’objet à copier.

| Destination Bucket | Spécifiez le nom du bucket de destination dans lequel l’objet sera copié.

| Destination Key | Spécifiez la nouvelle clé pour l’objet, après copie dans le bucket de destination.

| Server-Side Encryption | Cochez cette case pour activer le chiffrement côté serveur avec des clés de chiffrement gérées par Amazon S3 (SSE-S3) et utiliser le standard de chiffrement avancé 256 bits (AES-256) pour chiffrer vos données. Pour plus d’informations concernant le chiffrement côté serveur, consultez Protection des données à l’aide d’un chiffrement côté serveur.

| Use KMS | Cochez cette case pour activer le chiffrement côté serveur avec les clés gérées par AWS KMS (SSE-KMS) au lieu des clés de chiffrement gérées par Amazon S3 (SSE-S3). Cette propriété est disponible lorsque la case Server-Side Encryption est cochée.

| Customer Master Key | Spécifiez votre propre clé principale client (Customer Master Key, CMK) créée dans la console IAM à l’aide du service de gestion des clés AWS pour le chiffrement SSE-KMS. Si elle n’est pas spécifiée, la clé principale client par défaut, créée la première fois que vous ajoutez un objet chiffré SSE-KMS dans le bucket défini dans votre région, sera utilisée pour le chiffrement SSE-KMS. Cette propriété est disponible lorsque la case est Use KMS cochée.

| Die on error | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le traitement des lignes sans erreur.

**Advanced settings**

| Config client | Cochez cette case et spécifiez le(s) paramètre(s) client en cliquant sur le bouton [+] pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre client. Configurez la valeur des champs suivants pour chaque paramètre : |
• **Client Parameter**: cliquez dans la cellule et, dans la liste déroulante qui s’affiche, sélectionnez le paramètre client.

• **Value**: saisissez la valeur du paramètre sélectionné. Cette case est disponible uniquement lorsque la case *Use an existing connection* est décochée.

**STS Endpoint**

Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées. Cette case est disponible uniquement lorsque la case *Assume role* est cochée.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

**ERROR_MESSAGE**

Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case *Die on error* est cochée.

### Utilisation

**Règle d’utilisation**

Ce composant peut être utilisé en tant que composant standalone.

### Scénario : Copier un objet S3 d’un bucket à un autre

Ce scénario décrit un Job chargeant un nouvel objet dans un bucket S3 vide existant *bucket-src*, copiant l’objet du bucket *bucket-src* dans un autre bucket S3 vide *bucket-dst* puis listant l’objet dans le bucket *bucket-dst*, afin de voir si l’objet a bien été copié.
**Construire le Job**

**Procédure**

1. Créez un nouveau Job et ajoutez un composant `tS3Connection`, un `tS3Put`, un `tS3Copy`, un `tS3List`, un `tIterateToFlow` et un `tLogRow`, en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.

2. Reliez le `tS3List` au `tIterateToFlow` à l’aide d’un lien **Row > Iterate**.

3. Reliez le `tIterateToFlow` au `tLogRow` à l’aide d’un lien **Row > Main**.

4. Reliez le `tS3Connection` au `tS3Put` à l’aide d’un lien **Trigger > OnSubjobOk**.

5. Répétez l’opération pour relier le `tS3Put` au `tS3Copy` et le `tS3Copy` au `tS3List`.

**Configurer les composants**

**Créer une connexion à Amazon S3**

**Procédure**

1. Double-cliquez sur le `tS3Connection` pour ouvrir sa vue **Basic settings**.
2. Dans les champs **Access Key** et **Secret Key**, saisissez les informations d’authentification requises pour accéder à Amazon S3.


**Mettre à jour un objet dans un bucket Amazon S3**

**Procédure**

1. Double-cliquez sur le tS3Put pour ouvrir sa vue **Basic settings**.

   ![tS3Put_1](image1)

   2. Cochez la case **Use an existing connection** pour réutiliser les informations de connexion à Amazon S3 définies dans le composant tS3Connection.

   3. Dans le champ **Bucket**, saisissez le nom du bucket S3 dans lequel l’objet sera chargé. Dans cet exemple, le nom du bucket est *bucket-src*, il existe déjà dans Amazon S3.

   4. Dans le champ **Key**, saisissez la clé de l’objet à charger. Dans cet exemple, la clé est *tS3Copy_icon32_src.png*.

   5. Dans le champ **File**, parcourrez votre système ou saisissez le chemin à l’objet à charger. Dans cet exemple, saisissez *D:/tS3Copy_icon32.png*.

**Copier l’objet chargé dans un autre bucket Amazon S3**

**Procédure**

1. Double-cliquez sur le composant tS3Copy pour ouvrir sa vue **Basic settings**.

   ![tS3Copy_1](image2)

   2. Cochez la case **Use an existing connection** pour réutiliser les informations de connexion à Amazon S3 définies dans le composant tS3Connection.

   3. Dans le champ **Bucket**, dans la zone **Source Configuration**, saisissez le nom du bucket contenant l’objet à copier. Dans cet exemple, saisissez *bucket-src*. 
Dans le champ **Key**, dans la zone **Source Configuration**, saisissez la clé de l'objet à copier. Dans cet exemple, saisissez `tS3Copy_icon32_src.png`.

5. Dans le champ **Bucket**, dans la zone **Destination Configuration**, saisissez le nom du bucket dans lequel l'objet sera copié. Dans cet exemple, le bucket est le bucket vide `bucket-dst` existant dans Amazon S3.

6. Dans le champ **Key**, dans la zone **Destination Configuration**, saisissez la nouvelle clé de l'objet, après copie dans le bucket de destination. Dans cet exemple, saisissez `tS3Copy_icon32_dst.png`.

**Lister l'objet dans le bucket de destination**

**Procédure**

1. Double-cliquez sur le composant **tS3List** pour ouvrir sa vue **Basic settings**.

   ![tS3List_1](image)

   2. Cochez la case **Use an existing connection** pour réutiliser les informations de connexion à Amazon S3 définies dans le composant **tS3Connection**.

   3. Décochez la case **List all buckets objects**, puis cliquez sur le bouton `[+]` pour ajouter une ligne à la table **Bucket** affichée, puis configurez la valeur pour chaque colonne. Dans cet exemple, saisissez `bucket-dst` dans la colonne **Bucket name** et une valeur vide dans la colonne **Key prefix**. Ainsi, seuls les objets du bucket `bucket-dst` seront listés.

4. Double-cliquez sur le **tIterateToFlow** pour ouvrir sa vue **Basic settings**.

   ![tIterateToFlow_1](image)

   5. Cliquez sur le bouton `[...]` à côté du champ **Edit schema** et, dans l'éditeur du schéma, définissez le schéma en ajoutant une colonne **ObjectList** de type **String**.
6. Cliquez sur OK pour sauvegarder les modifications, puis cliquez sur Yes dans la boîte de dialogue qui s’ouvre, afin d’accepter la propagation.

7. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings.

8. Dans la zone Mode, sélectionnez Table (print values in cells of a table) pour une lisibilité optimale des résultats.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl + S pour sauvegarder le Job.
2. Appuyez sur F6 pour exécuter le Job.

```plaintext
[statistics] connecting to socket on port 3491
[statistics] connected
| tLogRow_1
| ObjectList
| tS3Copy_icon32_dat.png

[statistics] disconnected
```

Comme affiché ci-dessus, l’objet chargé a bien été copié dans le bucket de destination.
**tS3Delete**

Ce composant supprime un fichier dans Amazon S3.

**Propriétés du tS3Delete Standard**

Ces propriétés sont utilisées pour configurer le tS3Delete s'exécutant dans le framework de Jobs Standard.

Le composant tS3Delete Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Use existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Key</td>
<td>L’ID de la clé d’accès (Access Key) identifie uniquement un compte AWS. Pour plus d’informations concernant la clé d’accès et sa sécurité, consultez le site Obtention de vos clés d’accès AWS.</td>
</tr>
<tr>
<td>Secret Key</td>
<td>La Secret Access Key, combinée à la clé d’accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour obtenir les informations d’authentification de sécurité AWS depuis la métadonnée de l’instance Amazon EC2. Pour utiliser cette option, l’instance Amazon EC2 doit être démarrée et votre Job doit être en cours d’exécution sur Amazon EC2. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
</tbody>
</table>
| Assume role             | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  
  - **Role ARN** : nom Amazon Resource Name (ARN) du rôle.  
  - **Role session name** : identifiant de la session du rôle.  
  - **Session duration (minutes)** : durée (en minutes) pour laquelle est active la session du rôle.  
Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais). |
### Region
Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles ('us-east-1' par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Régions et points de terminaison AWS.

### Bucket
Nom du bucket, le dossier de plus haut niveau, dans Amazon S3.

### Key
Nom du fichier à supprimer.

### Die on error
Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.

#### Advanced settings

| Config client | Cochez cette case pour configurer des paramètres client.  
 Client parameter : sélectionnez les paramètres client dans la liste.  
 Value : saisissez la valeur des paramètres.  
Cette option n'est pas disponible lorsque la case Use existing connection est cochée. |
|---------------|--------------------------------------------------------------------------------------------------|
| STS Endpoint  | Cochez cette case et, dans le champ qui s'affiche, spécifiez l'endpoint du service AWS Security Token Service duquel les informations d'authentification sont récupérées.  
Cette case est disponible uniquement lorsque la case Assume role est cochée. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau des composants. |

#### Variables globales

| ERROR_MESSAGE | Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case Die on error est cochée. |

#### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant peut être utilisé en standalone ou avec d'autres composants S3, par exemple le tS3BucketList.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez</td>
</tr>
</tbody>
</table>
accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénario associé

Pour un scénario associé au composant tS3Delete, consultez Scénario : Vérifier l’absence d’un bucket, le créer et lister tous les buckets S3 à la page 3521.
Ce composant récupère un fichier d’Amazon S3.

**Propriétés du tS3Get Standard**

Ces propriétés sont utilisées pour configurer le tS3Get s’exécutant dans le framework de Jobs Standard.

Le composant tS3Get Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Access Key</strong></td>
<td>L’ID de la clé d’accès (Access Key) identifie uniquement un compte AWS. Pour plus d’informations concernant la clé d’accès et sa sécurité, consultez le site Obtention de vos clés d’accès AWS.</td>
</tr>
<tr>
<td><strong>Secret Key</strong></td>
<td>La Secret Access Key, combinée à la clé d’accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Inherit credentials from AWS role</strong></td>
<td>Cochez cette case pour obtenir les informations d’authentification de sécurité AWS depuis la métdonnée de l’instance Amazon EC2. Pour utiliser cette option, l’instance Amazon EC2 doit être démarrée et votre Job doit être en cours d’exécution sur Amazon EC2. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
<tr>
<td><strong>Assume role</strong></td>
<td>Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Role ARN</strong> : nom Amazon Resource Name (ARN) du rôle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Role session name</strong> : identifiant de la session du rôle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Session duration (minutes)</strong> : durée (en minutes) pour laquelle est active la session du rôle.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).</td>
</tr>
<tr>
<td>Region</td>
<td>Spécifiez la zone géographique AWS en sélectionnant le nom d'une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles (&quot;us-east-1&quot; par exemple) dans la liste. Pour plus d'informations sur les zones géographiques AWS, consultez Régions et points de terminaison AWS.</td>
</tr>
</tbody>
</table>
| Encrypt | Cochez cette case et dans la liste déroulante Key type affichée, sélectionnez l'une des trois options suivantes pour chiffrer les données côté client avant de les envoyer à Amazon S3. Pour plus d'informations, consultez Protection des données via le chiffrement côté client.  
  - **KMS-managed customer master key** : utilisez une clé maître client (CMK) gérée par KMS pour le chiffrrement des données côté client. Dans le champ Key, spécifiez l'identifiant de la clé maître client AWS gérée par KMS (CMK ID).  
  - **Symmetric Master Key** : utilisez une clé maître symétrique (clé secrète AES 256 bits) pour le chiffrrement des données côté client.  
    - **Algorithm** : sélectionnez l'algorithme associé à la clé dans la liste. Par défaut, il n'y a qu'un seul algorithme nommé AES.  
    - **Encoding** : sélectionnez le type de chiffrement associé à la clé dans la liste, Base64 ou X509.  
    - **Key ou Key file** : spécifiez la clé ou le chemin du fichier contenant la clé.  
  - **Asymmetric Master Key** : utilisez une clé maître asynétrique (une paire de clés RSA 1024 bits) pour le chiffrement des données côté client.  
    - **Algorithm** : sélectionnez l'algorithme associé à la clé dans la liste. Par défaut, il n'y a qu'un seul algorithme nommé RSA.  
    - **Public key file** : spécifiez le chemin du fichier contenant la clé publique.  
    - **Private key file** : spécifiez le chemin du fichier contenant la clé privée. |
| Bucket | Nom du bucket, le dossier de plus haut niveau, du serveur S3. |
| Key | Nom du fichier à sauvegarder sur le serveur S3. |
| File | Nom du fichier local à charger sur le serveur S3. |
| Die on error | Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. |
### Advanced settings

| Config client | Cochez cette case pour configurer des paramètres client.  
Client parameter : sélectionnez les paramètres client dans la liste.  
Value : saisissez la valeur des paramètres.  
Cette option n’est pas disponible lorsque la case Use existing connection est cochée. |
| --- | --- |
| STS Endpoint | Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.  
Cette case est disponible uniquement lorsque la case Assume role est cochée. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau des composants. |

### Variables globales

<table>
<thead>
<tr>
<th>ERROR_MESSAGE</th>
<th>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case Die on error est cochée.</th>
</tr>
</thead>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en stand alone ou avec d’autres composants S3, comme le tS3Connection.</th>
</tr>
</thead>
</table>
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  
Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.  
Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à |
Scénario associé

Pour un scénario associé utilisant le composant **tS3Get**, consultez **Scénario : Échange de fichiers avec Amazon S3** à la page 3558.
tS3List

Ce composant de liste les fichiers dans Amazon S3 selon les paramètres de préfixe du bucket ou du fichier.

**Propriétés du tS3List Standard**

Ces propriétés sont utilisées pour configurer le tS3List s'exécutant dans le framework de Jobs Standard.

Le composant tS3List Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Use existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Key</td>
<td>L’ID de la clé d’accès (Access Key) identifie uniquement un compte AWS. Pour plus d’informations concernant la clé d’accès et sa sécurité, consultez le site Obtention de vos clés d’accès AWS.</td>
</tr>
<tr>
<td>Secret Key</td>
<td>La Secret Access Key, combinée à la clé d’accès, constitue votre accès sécurisé à Amazon S3. Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour obtenir les informations d’authentification de sécurité AWS depuis la métadonnée de l’instance Amazon EC2. Pour utiliser cette option, l’instance Amazon EC2 doit être démarrée et votre Job doit être en cours d’exécution sur Amazon EC2. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2 .</td>
</tr>
</tbody>
</table>
| Assume role             | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  
  • Role ARN : nom Amazon Resource Name (ARN) du rôle.  
  • Role session name : identifiant de la session du rôle.  
  • Session duration (minutes) : durée (en minutes) pour laquelle est active la session du rôle.  
  Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais). |
### Region
Spécifiez la zone géographique AWS en sélectionnant le nom d’une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles ("us-east-1" par exemple) dans la liste. Pour plus d’informations sur les zones géographiques AWS, consultez Régions et points de terminaison AWS.

<table>
<thead>
<tr>
<th>List all bucket objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour lister tous les fichiers sur le serveur S3.</td>
</tr>
<tr>
<td><strong>Key prefix</strong> : saisissez le préfixe des fichiers à lister. Ainsi, seuls les fichiers contenant ce préfixe seront listés.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bucket</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cliquez sur le bouton [+] pour ajouter une ou plusieurs lignes afin de définir les préfixes des buckets et fichiers.</td>
</tr>
<tr>
<td><strong>Bucket name</strong> : nom du bucket, le dossier de plus haut niveau, duquel lister les fichiers.</td>
</tr>
<tr>
<td><strong>Key prefix</strong> : préfixe des fichiers à lister.</td>
</tr>
<tr>
<td>Cette option est indisponible lorsque la case List all bucket objects est cochée.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Die on error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>Config client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour configurer des paramètres client.</td>
</tr>
<tr>
<td><strong>Client parameter</strong> : sélectionnez les paramètres client dans la liste.</td>
</tr>
<tr>
<td><strong>Value</strong> : saisissez la valeur des paramètres.</td>
</tr>
<tr>
<td>Cette option n’est pas disponible lorsque la case Use existing connection est cochée.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STS Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.</td>
</tr>
<tr>
<td>Cette case est disponible uniquement lorsque la case Assume role est cochée.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour collecter les données de log au niveau des composants.</td>
</tr>
</tbody>
</table>

### Variables globales

<table>
<thead>
<tr>
<th>CURRENT_BUCKET_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nom du bucket courant. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CURRENT_KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clé courante. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>NB_BUCKET</strong></td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td><strong>NB_BUCKET_OBJECT</strong></td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
</tr>
</tbody>
</table>

### Utilisation

**Règle d’utilisation**

Ce composant peut être utilisé en standalone ou avec d’autres composants S3, comme le **tS3Delete**.

**Dynamic settings**

Cliquez sur le bouton `[+]` pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Scénario : Lister les fichiers d’un bucket ayant le même préfixe

Dans ce scénario, le **tS3List** est utilisé pour lister tous les fichiers d’un bucket ayant le même préfixe.

Voici les fichiers du bucket :

<table>
<thead>
<tr>
<th>Nom du fichier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fichier1</td>
<td>Fichier 1</td>
</tr>
<tr>
<td>fichier2</td>
<td>Fichier 2</td>
</tr>
<tr>
<td>fichier3</td>
<td>Fichier 3</td>
</tr>
</tbody>
</table>

...
Pour plus d’informations concernant la création d’un bucket et comment y mettre des fichiers, consultez Scénario : Vérifier l’absence d’un bucket, le créer et lister tous les buckets S3 à la page 3521 et Scénario : Échange de fichiers avec Amazon S3 à la page 3558.

**Construire le Job**

**Procédure**

1. De la Palette, déposez un composant tS3Connection, un tS3List, un tIterateToFlow, un tLogRow et un tS3Close dans l’espace de modélisation graphique.
2. Reliez le tS3Connection au tS3List à l’aide d’un lien OnSubjobOk.
3. Reliez le tS3List au tIterateToFlow à l’aide d’un lien Row > Iterate.
4. Reliez le tIterateToFlow au tLogRow à l’aide d’un lien Row > Main.
5. Reliez le tS3List au tS3Close à l’aide d’un lien OnSubjobOk.

**Configurer les composants**

**Procédure**

1. Double-cliquez sur le tS3Connection pour ouvrir sa vue Basic settings.
2. Dans les champs **Access Key** et **Secret Key**, saisissez les informations d’authentification.

3. Double-cliquez sur le **tS3List** pour ouvrir sa vue **Basic settings**.

4. Cochez la case **Use existing connection** afin de réutiliser la connexion définie précédemment.

5. Dans la table **Bucket**, cliquez sur le bouton [+] pour ajouter une ligne.

6. Dans les champs **Bucket name** et **Key prefix**, saisissez le nom du bucket et le préfixe du fichier. Ainsi, seuls les fichiers ayant le préfixe spécifié seront listés.

7. Double-cliquez sur le **tIterateToFlow** pour ouvrir sa vue **Basic settings**.

8. Cliquez sur **Edit schema** pour ouvrir l’éditeur du schéma.
Cliquez sur le bouton [+] pour ajouter une colonne. Nommez-la file_list et définissez son type à String.
Cliquez sur OK pour valider la configuration et fermer l’éditeur.

9. Dans la table Mapping, appuyez sur les touches Ctrl+Espace dans la colonne Value afin de sélectionner la variable tS3List_1_CURRENT_KEY.

10. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings.

Sélectionnez l’option Table (print values in cells of a table) pour un affichage optimal des résultats.

11. Double-cliquez sur le composant tS3Close afin d’ouvrir sa vue Basic settings.

Comme un seul composant de connexion est utilisé dans ce Job, vous n’avez pas besoin de le sélectionner dans la liste Component List, il est sélectionné par défaut.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl+S afin de sauvegarder votre Job.
2. Appuyez sur la touche F6 pour exécuter le Job.
Comme affiché ci-dessus, seuls les fichiers ayant le préfixe “in” sont listés.
**tS3Put**

Ce composant charge des données dans Amazon S3 depuis un fichier local ou depuis la mémoire cache, via le mode streaming.

**Propriétés du tS3Put Standard**

Ces propriétés sont utilisées pour configurer le tS3Put s’exécutant dans le framework de Jobs Standard.

Le composant tS3Put Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Access Key</strong></td>
<td>L’ID de la clé d’accès (Access Key) identifie uniquement un compte AWS. Pour plus d’informations concernant la clé d’accès et sa sécurité, consultez le site <strong>Obtention de vos clés d’accès AWS</strong>.</td>
</tr>
</tbody>
</table>
| **Secret Key**                              | La Secret Access Key, combinée à la clé d’accès, constitue votre accès sécurisé à Amazon S3.  

Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ **Secret key**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Inherit credentials from AWS role**      | Cochez cette case pour obtenir les informations d’authentification de sécurité AWS depuis la métadonnée de l’instance Amazon EC2. Pour utiliser cette option, l’instance Amazon EC2 doit être démarrée et votre Job doit être en cours d’exécution sur Amazon EC2. Pour plus d’informations, consultez **Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2**. |
| **Assume role**                             | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  

- **Role ARN** : nom Amazon Resource Name (ARN) du rôle.  

- **Role session name** : identifiant de la session du rôle.  

- **Session duration (minutes)** : durée (en minutes) pour laquelle est active la session du rôle.  

Pour plus d’informations concernant les rôles et AssumeRole, consultez **AssumeRole** (en anglais). |
| Region | Spécifiez la zone géographique AWS en sélectionnant le nom d’une zone géographique dans la liste ou en saisissant le nom de la zone entre guillemets doubles ("us-east-1" par exemple) dans la liste. Pour plus d’informations sur les zones géographiques AWS, consultez Régions et points de terminaison AWS. |
| Encrypt | Cochez cette case et dans la liste déroulante Key type affichée, sélectionnez l’une des trois options suivantes pour chiffrer les données côté client avant de les envoyer à Amazon S3. Pour plus d’informations, consultez Protection des données via le chiffrement côté client.  
  - **KMS-managed customer master key** : utilisez une clé maître client (CMK) gérée par KMS pour le chiffrement des données côté client. Dans le champ Key, spécifiez l’identifiant de la clé maître client AWS gérée par KMS (CMK ID).  
  - **Symmetric Master Key** : utilisez une clé maître symétrique (clé secrète AES 256 bits) pour le chiffrement des données côté client.  
    - **Algorithm** : sélectionnez l’algorithme associé à la clé dans la liste. Par défaut, il n’y a qu’un seul algorithme nommé AES.  
    - **Encoding** : sélectionnez le type de chiffrement associé à la clé dans la liste, Base64 ou X509.  
  - **Asymmetric Master Key** : utilisez une clé maître asymétrique (une paire de clés RSA 1024 bits) pour le chiffrement des données côté client.  
    - **Algorithm** : sélectionnez l’algorithme associé à la clé dans la liste. Par défaut, il n’y a qu’un seul algorithme nommé RSA.  
    - **Public key file** : spécifiez le chemin du fichier contenant la clé publique.  
    - **Private key file** : spécifiez le chemin du fichier contenant la clé privée. |
| Bucket | Nom du bucket, le dossier de plus haut niveau, du serveur S3. |
| Key | Nom du fichier qui contiendra les données chargées sur le serveur S3. |
| File or Stream | Chemin d’accès au fichier local à charger ou au flux d’entrée duquel les données seront chargées.  
Pour un exemple d’utilisation des flux, consultez Scénario 2 : Lire les données d’un fichier distant en mode stream à la page 1072. |
| Server-Side Encryption | Cochez cette case pour activer le chiffrement côté serveur avec des clés de chiffrement gérées par Amazon |
S3 (SSE-S3) et utiliser le standard de chiffrement avancé 256 bits (AES-256) pour chiffrer vos données.

Pour plus d’informations concernant le chiffrement côté serveur, consultez Protection des données à l’aide d’un chiffrement côté serveur.

<table>
<thead>
<tr>
<th>Use KMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour activer le chiffrement côté serveur avec les clés gérées par AWS KMS (SSE-KMS) au lieu des clés de chiffrement gérées par Amazon S3 (SSE-S3).</td>
</tr>
<tr>
<td>Cette propriété est disponible lorsque la case Server-Side Encryption est cochée.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customer Master Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spécifiez votre propre clé principale client (Customer Master Key, CMK) créée dans la console IAM à l’aide du service de gestion des clés AWS pour le chiffrement SSE-KMS. Si elle n’est pas spécifiée, la clé principale client par défaut, créée la première fois que vous ajoutez un objet chiffré SSE-KMS dans le bucket défini dans votre région, sera utilisée pour le chiffrement SSE-KMS.</td>
</tr>
<tr>
<td>Cette propriété est disponible lorsque la case est Use KMS cochée.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Die on error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cette case est décochée par défaut et permet de terminer le traitement avec les lignes sans erreur, ainsi que d’ignorer les lignes en erreur.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>Config client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour configurer des paramètres client.</td>
</tr>
<tr>
<td><strong>Client parameter</strong> : sélectionnez les paramètres client dans la liste.</td>
</tr>
<tr>
<td><strong>Value</strong> : saisissez la valeur des paramètres.</td>
</tr>
<tr>
<td>Cette option n’est pas disponible lorsque la case Use existing connection est cochée.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STS Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.</td>
</tr>
<tr>
<td>Cette case est disponible uniquement lorsque la case Assume role est cochée.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Part size (Mb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spécifiez la taille de la partition en Mo lors du chargement d’un fichier volumineux partitionné.</td>
</tr>
<tr>
<td>Pour plus d’informations concernant le chargement partitionné de S3, consultez Présentation du téléchargement partitionné.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour collecter les données de log au niveau des composants.</td>
</tr>
</tbody>
</table>
Variables globales

| **ERROR_MESSAGE** | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case **Die on error** est cochée. |

Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en standalone ou avec d’autres composants S3, par exemple le **tS3Connection**. |
| Dynamic settings | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

Scénario : Échange de fichiers avec Amazon S3

Dans ce scénario, le composant **tS3Put** est utilisé pour envoyer un fichier sur le serveur Amazon S3 pendant que le **tS3Get** récupère ce fichier.

Construire le Job

Procédure

1. Déposez un **tS3Connection**, un **tS3Put** et un **tS3Get** de la Palette dans l’espace de modélisation graphique.
2. Reliez le **tS3Connection** au **tS3Put** à l’aide d’un lien **OnSubjobOk**.
3. Reliez le **tS3Put** au **tS3Get** à l’aide d’un lien **OnSubjobOk**.
Configurer les composants

Procédure

1. Double-cliquez sur le **tS3Connection** pour ouvrir sa vue **Basic settings**.

2. Dans les champs **Access Key** et **Secret Key**, saisissez les informations d’authentification.

3. Double-cliquez sur le **tS3Put** pour ouvrir sa vue **Basic settings**.

4. Cochez la case **Use existing connection** pour réutiliser la connexion.

5. Dans le champ **Bucket**, saisissez le nom du bucket.

6. Dans le champ **Key**, saisissez le nom du fichier à sauvegarder sur le serveur S3.

7. Dans le champ **File**, saisissez le nom du fichier à charger dans le serveur S3.

8. Double-cliquez sur le composant **tS3Get** pour ouvrir sa vue **Basic settings**.
9. Cochez la case **Use existing connection** pour réutiliser la connexion.

10. Dans le champ **Bucket**, saisissez le nom du bucket.

11. Dans le champ **Key**, saisissez le nom du fichier à récupérer du serveur S3.

12. Dans le champ **File**, saisissez le nom local du fichier récupéré.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.

2. Appuyez sur **F6** pour exécuter le Job.

   ```
   Starting job s3_put_get at 14:22 08/10/2013.
   [statistics] connecting to socket on port 3950
   [statistics] connected
   [statistics] disconnected
   Job s3_put_get ended at 14:22 08/10/2013. [exit code=0]
   ```

3. Consultez la console Web S3 :

   ![Console Web S3](https://console.aws.amazon.com/s3/home?n)

   Comme affiché ci-dessus, le fichier a bien été mis sur le serveur S3.

4. Rafraîchissez le client S3 Browser :
Comme affiché ci-dessus, cela montre également l’action réussie du composant S3 Put.

5. Allez dans le dossier où est sauvegardé localement le fichier récupéré :

Comme affiché ci-dessus, le fichier distant est récupéré dans le disque local, ce qui prouve que l’action Get S3 s’est bien déroulée.
tSageX3Input

Ce composant extrait des données d'un système Sage X3 donné.
Le tSageX3Input tire profit du Service Web fourni par un serveur Web Sage X3 pour extraire des données du système Sage X3 (serveur X3).

**Propriétés du tSageX3Input Standard**

Ces propriétés sont utilisées pour configurer le tSageX3Input s'exécutant dans le framework de Jobs Standard.
Le composant tSageX3Input Standard appartient à la famille Business.
Le composant de ce framework est toujours disponible.

**Basic settings**

| **Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

| **Built-in** | Propriétés utilisées ponctuellement. |
| **Repository** | Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées. |
| **Endpoint address** | Saisissez l'adresse du Service Web fourni par le serveur Web Sage X3 donné. |
| **Username et Password** | Saisissez les informations d'authentification pour le Service Web, définies lors de la configuration du Serveur Web Sage X3. |
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Language**
Saisissez le nom du code langue X3 utilisé pour le démarrage du groupe de connexions.

**Pool alias**
Saisissez le nom du pool de connexion distribuant les requêtes reçues aux connexions disponibles. Ce nom est donné depuis la console de configuration Sage X3.

**Request config**
Saisissez la chaîne de caractères de configuration si vous souhaitez récupérer les informations de débogage ou de traces. Par exemple, la chaîne de caractères peut être :

```plaintext
RequestConfigDebug="adwxss.trace.on=on";
```

Si vous devez utiliser plusieurs chaînes de caractères, séparez-les avec un `&`, par exemple,

```plaintext
RequestConfigDebug="adwxss.trace.on=on&adwxss.trace.size=16384";
```

⚠️ **Avertissement** :
Un outil tiers est nécessaire pour récupérer ce type d'informations.

**Publication name**
Saisissez dans ce champ le nom de l'objet, de la liste ou du sous-programme publié(e) auquel/à laquelle vous souhaitez que votre **Studio Talend** accède.

**Action**
Sélectionnez dans la liste l'action à effectuer.
- **getAllDetails** : retourne les détails de tous les enregistrements.
- **getSummary** : retourne le résumé de chaque enregistrement.
- **getDetail** : retourne les détails des enregistrements répondant aux conditions configurées dans la table **Query condition**.

**Mapping**
Remplissez ce tableau pour mapper les éléments variables de l’objet, le sous-programme ou la liste défini(e)(s) dans le serveur Web Sage X3 donné. Les colonnes à compléter incluent :

- **Column** : les colonnes définies dans l’éditeur de schéma de ce composant.
- **Group ID** : l’identifiant de chaque groupe d’éléments variables. Par exemple, un groupe d’éléments variables peut représenter l’un des attributs d’un objet.
- **Field name** : le nom de chaque élément variable.

**Conditions**
Renseignez cette table afin de configurer la (les) condition(s) de requête. Les colonnes à renseigner comprennent :
<table>
<thead>
<tr>
<th><strong>Key</strong> : les noms des éléments de variables utilisés comme clé pour l'extraction de données.</th>
<th><strong>Value</strong> : la valeur du champ de clé donné utilisé pour extraire les données correspondantes. Disponible lorsque l’option <strong>getDetail</strong> est sélectionnée dans la liste <strong>Action</strong>.</th>
</tr>
</thead>
</table>
| **Query condition** | Cochez cette case pour paramétrer la (les) condition(s) de la requête. Les colonnes à compléter incluent :

- **Key** : le nom des éléments variables utilisés comme clé de l'extraction de données.
- **Value** : la valeur du champ clé donné utilisé pour extraire les données correspondantes.
Indisponible lorsque l’option **getDetail** est sélectionnée dans la liste **Action**. |
| **Limit** | Saisissez un chiffre ou un nombre indiquant le maximum de lignes de données à extraire.
Indisponible lorsque l’option **getDetail** est sélectionnée dans la liste **Action**. |

**Advanced settings**

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

**Global Variables**

| **Global Variables** | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| **Règle d’utilisation** | Généralement utilisé comme composant de début. Un composant de sortie est nécessaire. |
Scénario : Utiliser les clés des requêtes pour extraire des données d'un système Sage X3 donné

Ce scénario décrit un Job à deux composants utilisé pour extraire une ligne ou des données d'un système Sage X3 donné. La méthode des objets est appelée, les éléments variables de cet objet sont des attributs. Les données utilisées dans ce scénario proviennent de l'exemple fourni par Sage X3.

Configurer le Job

Procédure

1. Déposez un composant tSageX3Input et un tLogRow de la Palette dans l'espace de modélisation graphique.
2. Reliez le composant tSageX3Input au tLogRow à l'aide d'un lien Row > Main.

Configurer le schéma du tSageX3Input

Procédure

1. Double-cliquez sur le tSageX3Input afin de configurer ses Basic settings dans la vue Component.
2. Dans la liste **Schema**, sélectionnez **Built-In** puis cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l’éditeur du schéma.

3. Dans cet éditeur, cliquez douze fois sur le bouton [+] pour ajouter 12 lignes à la table.

5. Dans la colonne Type, cliquez sur la ligne IMG pour afficher une liste déroulante.


7. Cliquez sur OK pour valider les modifications et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

**Configurer la connexion au serveur Web Sage X3**

**Procédure**


2. Dans le champ User, saisissez le nom de l’utilisateur de Sage X3. Dans cet exemple, l’utilisateur est ERP.

3. Dans le champ Language, saisissez le nom du code langue X3 utilisé pour le démarrage d’un groupe de connexions, FRA dans cet exemple.

4. Dans le champ Pool alias, saisissez le nom du pool de connexion à utiliser. Dans cet exemple, le pool de connexion est appelé TALEND.

5. Dans le champ Publication name, saisissez le nom de publication de l’objet à appeler. Dans ce scénario, le nom de publication est ITMDET.

**Paramétrer le mapping et configurer la condition de la requête**

**Procédure**

1. Dans les colonnes Group ID et Field name de la table Mapping, saisissez les valeurs correspondant à l’ID du groupe d’attributs et au nom de publication définis dans le serveur Web Sage X3. Dans cet exemple, les valeurs sont présentées comme suit :

<table>
<thead>
<tr>
<th>Column</th>
<th>Group ID</th>
<th>Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITMREF</td>
<td>&quot;TJT0_1&quot;</td>
<td>&quot;TITMREF&quot;</td>
</tr>
<tr>
<td>ZITMREF</td>
<td>&quot;TJT0_1&quot;</td>
<td>&quot;ZITMREF&quot;</td>
</tr>
<tr>
<td>INTDES1</td>
<td>&quot;TJT0_1&quot;</td>
<td>&quot;INTDES1&quot;</td>
</tr>
<tr>
<td>ENAPLG</td>
<td>&quot;TJT0_1&quot;</td>
<td>&quot;ENAPLG&quot;</td>
</tr>
<tr>
<td>VIG</td>
<td>&quot;TJT3_1&quot;</td>
<td>&quot;VIG&quot;</td>
</tr>
<tr>
<td>IMG</td>
<td>&quot;TJT3_1&quot;</td>
<td>&quot;IMG&quot;</td>
</tr>
<tr>
<td>TIT2NBLIG</td>
<td>&quot;TJT2_1&quot;</td>
<td>&quot;TIT2-NBLIG&quot;</td>
</tr>
<tr>
<td>ITMLNK</td>
<td>&quot;TJT2_1&quot;</td>
<td>&quot;ITMLNK&quot;</td>
</tr>
<tr>
<td>ZITMLNK</td>
<td>&quot;TJT2_1&quot;</td>
<td>&quot;ZITMLNK&quot;</td>
</tr>
<tr>
<td>TEXTE</td>
<td>&quot;TJT3_2&quot;</td>
<td>&quot;TEXTE&quot;</td>
</tr>
<tr>
<td>WWW_MODSTAMP</td>
<td>&quot;ADXTEC&quot;</td>
<td>&quot;WWW_MODSTAMP&quot;</td>
</tr>
<tr>
<td>WWW_MODUSER</td>
<td>&quot;ADXTEC&quot;</td>
<td>&quot;WWW_MODUSER&quot;</td>
</tr>
</tbody>
</table>

**Remarque :**

Dans la table Mapping, la colonne Column a été automatiquement renseignée avec les colonnes que vous avez créées dans l’éditeur du schéma.
2. Cochez la case **Query condition** pour activer le tableau **Conditions**.
3. Sous le tableau **Conditions**, cliquez sur le bouton \([+]\) pour ajouter une ligne.
4. Dans la colonne **Key**, saisissez le nom de publication associé à l'attribut de l'objet dont vous souhaitez extraire les données.
5. Dans la colonne **Value**, saisissez la valeur de l'attribut que vous avez sélectionné comme clé de l'extraction. Dans ce scénario, sa valeur est **CONTS00059**, une des références produit.

**Exécuter le Job**

**Procédure**

1. Sélectionnez **Built-In** dans la liste **Schema** puis cliquez sur \([...]\) à côté du champ **Edit schema** afin d'ouvrir l'éditeur du schéma.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TMPREF&quot;</td>
<td>&quot;CONTS00059&quot;</td>
</tr>
</tbody>
</table>

2. Appuyez sur **F6** pour exécuter le Job.
   Les résultats sont affichés dans la vue **Run** :

   ![Résultats](image)
**tSageX3Output**

Ce composant écrit des données dans un système Sage X3 donné.

Le tSageX3Output se connecte au service Web fourni par un serveur Web Sage X3 donné et à partir de là, insère, met à jour ou supprime des données dans le système Sage X3 (serveur X3).

**Propriétés du tSageX3Output Standard**

Ces propriétés sont utilisées pour configurer le tSageX3Output s’exécutant dans le framework de Jobs Standard.

Le composant tSageX3Output Standard appartient à la famille Business.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode [Repository], trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode [Built-In] et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

<table>
<thead>
<tr>
<th>Built-in</th>
<th>Propriétés utilisées ponctuellement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier où sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Endpoint address</td>
<td>Saisissez l’adresse du Service Web fourni par le serveur Web Sage X3 donné.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez les informations d’authentification pour le Service Web, définies lors de la configuration du Serveur Web Sage X3. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Language</strong></td>
<td>Saisissez le nom du code langue X3 utilisé pour le démarrage du groupe de connexions.</td>
</tr>
<tr>
<td><strong>Pool alias</strong></td>
<td>Saisissez le nom du pool de connexion distribuant les requêtes reçues aux connexions disponibles. Ce nom est donné depuis la console de configuration Sage X3.</td>
</tr>
</tbody>
</table>
| **Request config**      | Saisissez la chaîne de caractères de configuration si vous souhaitez récupérer les informations de débogage ou de traces. Par exemple, la chaîne de caractères peut être : 

```
RequestConfigDebug="adxwss.trace.on=on";
```

Si vous devez utiliser plusieurs chaînes de caractères, séparez-les avec un & par exemple,

```
RequestConfigDebug="adxwss.trace.on=on&adxwss.trace.size=16384";
```

**Avertissement :**

Un outil tiers est nécessaire pour récupérer ce type d’informations.

<table>
<thead>
<tr>
<th><strong>Publication name</strong></th>
<th>Saisissez dans ce champ le nom de l’objet, de la liste ou du sous-programme publié(e) auquel/à laquelle vous souhaitez que votre <strong>Studio Talend</strong> accède.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action</strong></td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données d’un système Sage X3:</td>
</tr>
<tr>
<td></td>
<td><strong>Insert</strong> : insérer des données,</td>
</tr>
<tr>
<td></td>
<td><strong>Update</strong> : mettre à jour des données,</td>
</tr>
<tr>
<td></td>
<td><strong>Delete</strong> : supprimer des données.</td>
</tr>
<tr>
<td><strong>Mapping</strong></td>
<td>Remplissez ce tableau pour mapper les éléments variables de l’objet, le sous-programme ou la liste défini(e)(s) dans le serveur Web Sage X3 donné au(x)quel(s)/à laquelle votre <strong>Studio Talend</strong> accède. Seuls les éléments sur lesquels vous souhaitez effectuer l’action sont sélectionnés et saisis dans le but de réaliser le mapping. Les colonnes à compléter incluent :</td>
</tr>
<tr>
<td></td>
<td><strong>Column</strong> : les colonnes définies dans l’éditeur du schéma de ce composant.</td>
</tr>
<tr>
<td></td>
<td><strong>Key</strong> : l’élément variable utilisé comme clé de l’insertion, de la mise à jour ou de la suppression de données. Cochez la case correspondante si la clé est un élément variable.</td>
</tr>
</tbody>
</table>
**Group ID**: l'identifiant de chaque groupe d'éléments variables. Par exemple, un groupe d'éléments variables peut représenter l'un des attributs d'un objet.

**Field name**: le nom du champ de chaque élément variable sélectionné.

### Advanced settings

| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant. |

### Global Variables

| **Global Variables** | **ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

### Utilisation

| **Règle d’utilisation** | Ce composant est généralement utilisé comme composant de sortie et requiert un composant d’entrée. |

### Scénario : Utiliser un service Web Sage X3 pour insérer des données dans un système Sage X3 donné

Ce scénario décrit un Job à deux composants, utilisé pour générer une ligne de données et insérer ces données dans un système Sage X3 donné. Vous pouvez trouver les données utilisées dans ce scénario dans l’exemple fourni par Sage X3. Le service Web Sage X3 est utilisé pour accéder à un objet.
Configurer le Job

Procédure

1. Déposez un composant *tFixedFlowInput* et un *tSageX3Output* de la *Palette* dans l'espace de modélisation graphique.
2. Reliez le composant *tFixedFlowInput* au composant *tSageX3Output* à l'aide d'un lien *Row > Main*.

Configurer le schéma des données d'entrée

Procédure

1. Double-cliquez sur le *tFixedFlowInput* afin d'afficher sa vue *Component* et définir ses propriétés.
2. Cliquez sur le bouton [...] à côté du champ *Edit schema* afin d'ouvrir l'éditeur du schéma.
3. Cliquez quatre fois sur le bouton [+] afin d'ajouter quatre lignes au schéma.
4. Cliquez sur **OK** pour valider les modifications, puis acceptez la propagation proposée par la boîte de dialogue qui s’ouvre ensuite. Les quatre lignes apparaissent automatiquement dans la table **Values** de la vue **Component**.

5. Dans le tableau **Values** de la zone **Mode**, saisissez les valeurs de chacune des quatre lignes de la colonne **Value**. Dans ce scénario, les valeurs (de haut en bas) sont : CONTS00059, Screen 24\" standard 16/10, Screen 24\" standard 28/10 et 2.

**Résultats**

⚠ Avertissement :

Les valeurs dans la colonne **Value** doivent être mises entre guillemets.

**Configurer la connexion au serveur Web Sage X3**

**Procédure**

1. Double-cliquez sur le composant **tSageX3Output** afin d’afficher sa vue **Component** et définir ses propriétés.


3. Dans le champ **User**, saisissez le nom de l’utilisateur de Sage X3. Dans cet exemple, l’utilisateur est **ERP**.

4. Dans le champ **Language**, saisissez le nom du code langue X3 utilisé pour démarrer un groupe de connexions. Dans cet exemple, le code est **FRA**.

5. Dans le champ **Pool alias**, saisissez le nom du pool de connexion à utiliser. Dans cet exemple, le pool de connexion est appelé **TALEND**.

6. Dans le champ **Publication name**, saisissez le nom de publication de l’objet à appeler. Dans ce scénario, le nom de publication est **ITMDET**.

Paramétrer le mapping

Procédure

1. Dans la colonne **Field name** de la table **Mapping**, saisissez le nom des champs des attributs sur lesquels effectuer l’action sélectionnée.

2. Dans la colonne **Group ID** de la table **Mapping**, saisissez les valeurs correspondantes afin de regrouper les IDs des attributs sélectionnés. Ces IDs sont définis dans le serveur Web Sage X3.

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Group ID</th>
<th>Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>newColumn</td>
<td></td>
<td>&quot;TITO_1&quot;</td>
<td>&quot;TIMREF&quot;</td>
</tr>
<tr>
<td>newColumn1</td>
<td></td>
<td>&quot;TITO_1&quot;</td>
<td>&quot;ZITMREF&quot;</td>
</tr>
<tr>
<td>newColumn2</td>
<td></td>
<td>&quot;TITO_1&quot;</td>
<td>&quot;INTDE51&quot;</td>
</tr>
<tr>
<td>newColumn3</td>
<td></td>
<td>&quot;TITO_1&quot;</td>
<td>&quot;EIAFLC&quot;</td>
</tr>
</tbody>
</table>

Remarque :

Dans la table **Mapping**, la colonne **Column** a été automatiquement renseignée avec toutes les colonnes récupérées du schéma du composant précédent.

Exécuter le Job

Appuyez sur **Ctrl+Espace** afin de sauvegarder votre Job puis sur **F6** pour exécuter le Job.

Pour vérifier les données que vous avez insérées dans ce scénario, vous pouvez utiliser le composant **tSageX3Input** pour lire les données concernées dans le serveur Sage X3.

Pour plus d’informations concernant l’utilisation du composant **tSageX3Input** pour lire des données, consultez le composant **tSageX3Input**.
**tSalesforceBulkExec**

Ce composant charge en masse des données dans un fichier donné dans un objet Salesforce.

Les composants tSalesforceOutputBulk et tSalesforceBulkExec sont utilisés ensemble dans un processus de deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la seconde, ce fichier est utilisé pour alimenter la base de données Salesforce. Ces deux étapes sont fusionnées dans le composant tSalesforceOutputBulkExec. L’avantage d’utiliser deux étapes séparées réside dans le fait que les données peuvent être transformées avant chargement dans la base de données.

**Propriétés du tSalesforceBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tSalesforceBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tSalesforceBulkExec Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spéciﬁer manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.

<table>
<thead>
<tr>
<th>Connection Component</th>
<th>Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Sélectionnez dans la liste déroulante le type de connexion. Les propriétés de la connexion varient selon le type de connexion sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Basic</strong> : sélectionnez cette option pour accéder à Salesforce en saisissant votre identifiant et votre mot de passe.</td>
</tr>
<tr>
<td></td>
<td>• <strong>OAuth</strong> : sélectionnez cette option pour accéder à Salesforce via le protocole OAuth (Open Authorization) 2.0 pour authentifier l’utilisateur. Dans la liste déroulante OAuth2 ﬂow type qui s’affiche, sélectionnez un type de ﬂux d’autentiﬁcation OAuth 2.0, Json Web Token Flow ou Implicit</td>
</tr>
</tbody>
</table>
Flow (Deprecated) (User Agent Flow). Ainsi, vous devez créer une application connectée dans Salesforce, afin de configurer un fournisseur d’authentification Salesforce.

Pour plus d’informations concernant le flux d’authentification OAuth, consultez Authenticate Apps with OAuth (en anglais).

Pour plus d’informations concernant la création d’une application connectée dans Salesforce, consultez Create a Connected App (en anglais).

<table>
<thead>
<tr>
<th><strong>User Id</strong></th>
<th>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Password</strong></td>
<td>Mot de passe Salesforce associé à l’identifiant. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</td>
</tr>
<tr>
<td><strong>Security Token</strong></td>
<td>Jeton de sécurité Salesforce. Pour plus d’informations, consultez Reset Your Security Token (en anglais). Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</td>
</tr>
<tr>
<td><strong>Issuer</strong></td>
<td>Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez Create a Connected App (en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Subject</strong></td>
<td>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Expiration time (in seconds)</strong></td>
<td>Délai d’expiration de l’assertion (en secondes) durant les cinq prochains minutes. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Key store</strong></td>
<td>Chemin pointant vers le fichier Keystore au format Java Keystore (JKS). Le fichier Keystore peut être généré en créant un certificat signé par Salesforce et en l’exportant vers le Keystore. Pour plus d’informations, consultez Generate a Self-Signed Certificate (en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Key store password</strong></td>
<td>Mot de passe du Keystore. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Client Id</td>
<td>Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez Create a Connected App (en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td>Callback Host</td>
<td>Valeur de l’hôte dans l’URL de la fonction de rappel pour l’authentification OAuth définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td>Callback Port</td>
<td>Valeur du port dans l’URL de la fonction de rappel pour l’authentification OAuth, définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td>Token File</td>
<td>Chemin d’accès au fichier de jeton stockant le jeton de rafraîchissement utilisé pour obtenir le jeton d’accès sans autorisation. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td>Module Name</td>
<td>Cliquez sur le bouton [...] à côté du champ et, dans la boîte de dialogue qui s’ouvre, sélectionnez le module à utiliser, ou cochez la case Use custom object et spécifiez le nom du module dans le champ Object Name.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
</tbody>
</table>
- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

### Output Action

Sélectionnez dans la liste l’une des opérations suivantes à effectuer.
- **INSERT**: insérer un ou plusieurs enregistrements dans Salesforce.
- **UPDATE**: mettre à jour un ou plusieurs enregistrements existants dans Salesforce.
- **UPSERT**: créer de nouveaux enregistrements et mettre à jour des enregistrements existants. Dans le champ **Upsert Key Column** affiché, spécifiez la colonne clé pour l’opération d’upsert.
- **DELETE**: supprimer un ou plusieurs enregistrements dans Salesforce.

### Bulk File Path

Spécifiez le chemin d’accès au fichier stockant les données à traiter.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Salesforce URL</strong></th>
<th>URL de Service Web, requise pour vous connecter à Salesforce.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>API version</strong></td>
<td>Version de l'API Salesforce.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété est disponible uniquement lorsque le type de connexion <strong>OAuth</strong> est sélectionné.</td>
</tr>
<tr>
<td><strong>Need compression</strong></td>
<td>Cochez cette case pour activer la compression des messages SOAP, ce qui peut conduire à améliorer les performances.</td>
</tr>
<tr>
<td><strong>Trace HTTP message</strong></td>
<td>Cochez cette case pour écrire en sortie les interactions HTTP dans la console.</td>
</tr>
<tr>
<td><strong>Client Id</strong></td>
<td>Saisissez l’ID d’un utilisateur réel afin de différencier les utilisateurs d’un même compte et mot de passe pour accéder au site Web de Salesforce.</td>
</tr>
</tbody>
</table>
Timeout | Saisissez, en millisecondes, le délai avant suspension des requêtes dans Salesforce.

Use Proxy | Cochez cette case pour utiliser un serveur proxy et, dans les champs Host, Port, User Id et Password qui s’affichent, spécifiez les paramètres de connexion du serveur proxy.

Concurrency Mode | Sélectionnez le mode de simultanéité du Job.
- **Parallel** : traite les lots en mode parallèle.
- **Serial** : traite les lots en mode séquentiel.

Rows to Commit | Spécifiez le nombre de lignes par lot de données à traiter.

Bytes to Commit | Spécifiez le nombre d'octets par lot de données à traiter.

Wait Time Check Batch State | Spécifiez le temps d'attente (en millisecondes) avant de vérifier que les lots dans un Job ont été traités, jusqu'à ce que tous les lots soient traités.

tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.

**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_SUCCESS</td>
<td>Nombre de lignes traitées correctement. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_REJECT</td>
<td>Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec le **tSalesforceOutputBulk**. Utilisés ensemble, ils permettent un gain de performance lors de l’alimentation ou de la modification d’informations dans Salesforce. |

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tSalesforceConnection

Ce composant ouvre une connexion à Salesforce.

Propriétés du tSalesforceConnection Standard

Ces propriétés sont utilisées pour configurer le tSalesforceConnection s’exécutant dans le framework de Jobs Standard.

Le composant tSalesforceConnection Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Sélectionnez dans la liste déroulante le type de connexion. Les propriétés de la connexion varient selon le type de connexion sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Basic</strong> : sélectionnez cette option pour accéder à Salesforce en saisissant votre identifiant et votre mot de passe.</td>
</tr>
</tbody>
</table>
| **User Id** | Identifiant Salesforce.  
Cette propriété est disponible uniquement lorsque le type de connexion **Basic** est sélectionné. |
| **Password** | Mot de passe Salesforce associé à l’identifiant.  
Cette propriété est disponible uniquement lorsque le type de connexion **Basic** est sélectionné. |
| **Security Token** | Jeton de sécurité Salesforce. Pour plus d’informations, consultez **Reset Your Security Token** (en anglais).  
Cette propriété est disponible uniquement lorsque le type de connexion **Basic** est sélectionné. |
| **Issuer** | Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez **Create a Connected App** (en anglais).  
Cette propriété est disponible uniquement lorsque le type OAuth **Json Web Token Flow** est sélectionné. |
| **Subject** | Identifiant Salesforce.  
Cette propriété est disponible uniquement lorsque le type OAuth **Json Web Token Flow** est sélectionné. |
| **Expiration time (in seconds)** | Délai d’expiration de l’assertion (en secondes) durant les cinq prochains minutes.  
Cette propriété est disponible uniquement lorsque le type OAuth **Json Web Token Flow** est sélectionné. |
| **Key store** | Chemin pointant vers le fichier Keystore au format Java Keystore (JKS).  
Le fichier Keystore peut être généré en créant un certificat signé par Salesforce et en l’exportant vers le Keystore. Pour plus d’informations, consultez **Generate a Self-Signed Certificate** (en anglais).  
Cette propriété est disponible uniquement lorsque le type OAuth **Json Web Token Flow** est sélectionné. |
| **Key store password** | Mot de passe du Keystore.  
Cette propriété est disponible uniquement lorsque le type OAuth **Json Web Token Flow** est sélectionné. |
| **Certificate alias** | Nom unique du certificat signé par Salesforce.  
Cette propriété est disponible uniquement lorsque le type OAuth **Json Web Token Flow** est sélectionné. |
| **Client Id** | Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez **Create a Connected App** (en anglais). |
Cette propriété est disponible uniquement lorsque le type OAuth **Implicit Flow** est sélectionné.

### Client Secret
Consumer Secret d’OAuth, généré lorsque votre application connectée est créée et affichée dans la page des informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez [Create a Connected App](en anglais).

Cette propriété est disponible uniquement lorsque le type OAuth **Implicit Flow** est sélectionné.

### Callback Host
Valeur de l’hôte dans l’URL de la fonction de rappel pour l’authentification OAuth définie lors de la création d’une application connectée et qui sera affichée dans la zone **API (Enable OAuth Settings)** de la page d’informations de l’application connectée dans Salesforce.

Cette propriété est disponible uniquement lorsque le type OAuth **Implicit Flow** est sélectionné.

### Callback Port
Valeur du port dans l’URL de la fonction de rappel pour l’authentification OAuth, définie lors de la création d’une application connectée et qui sera affichée dans la zone **API (Enable OAuth Settings)** de la page d’informations de l’application connectée dans Salesforce.

Cette propriété est disponible uniquement lorsque le type OAuth **Implicit Flow** est sélectionné.

### Token File
Chemin d’accès au fichier de jeton stockant le jeton de rafraîchissement utilisé pour obtenir le jeton d’accès sans autorisation.

Cette propriété est disponible uniquement lorsque le type OAuth **Implicit Flow** est sélectionné.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Salesforce URL</strong></th>
<th>URL de Service Web, requise pour vous connecter à Salesforce.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>API version</strong></td>
<td>Version de l’API Salesforce.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété est disponible uniquement lorsque le type de connexion OAuth est sélectionné.</td>
</tr>
<tr>
<td><strong>Bulk Connection</strong></td>
<td>Cochez cette case si vous souhaitez utiliser la fonction de traitement de masse de données.</td>
</tr>
<tr>
<td><strong>Use or save the connection session</strong></td>
<td>Cochez cette case et, dans le champ <strong>Session directory</strong> qui s’affiche, spécifiez le chemin vers le fichier de session de connexion à sauvegarder ou utiliser. Ce fichier de session peut être partagé par différents Jobs afin de récupérer une session de connexion, tant que le bon ID utilisateur est fourni par le composant. Ainsi, vous n’avez pas besoin de vous connecter au serveur pour récupérer la session. Lorsqu’une session expirée est détectée, si les informations de connexion sont fournies (ID utilisateur,</td>
</tr>
</tbody>
</table>
mot de passe et clé de sécurité) le composant va se connecter au serveur pour récupérer les informations de la nouvelle session et mettre à jour le fichier de session de connexion.

Cette propriété est disponible uniquement lorsque le type de connexion **Basic** est sélectionné.

<table>
<thead>
<tr>
<th>Need compression</th>
<th>Cochez cette case pour activer la compression des messages SOAP, ce qui peut conduire à améliorer les performances.</th>
</tr>
</thead>
</table>
| Use Http Chunked | Cochez cette case pour utiliser le mécanisme de transfert des données HTTP morcelées. 
Cette propriété est disponible uniquement lorsque la case **Bulk Connection** est décochée. |
| Trace HTTP message | Cochez cette case pour écrire en sortie les interactions HTTP dans la console. 
Cette propriété est disponible uniquement lorsque la case **Bulk Connection** est cochée. |
| Client Id | Saisissez l’ID d’un utilisateur réel afin de différencier les utilisateurs d’un même compte et mot de passe pour accéder au site Web de Salesforce. |
| Timeout | Saisissez, en millisecondes, le délai avant suspension des requêtes dans Salesforce. |
| Use Proxy | Cochez cette case pour utiliser un serveur proxy et, dans les champs **Host**, **Port**, **User Id** et **Password** qui s’affiche, spécifiez les paramètres de connexion du serveur proxy. |
| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec les composants Salesforce. |

**Se connecter à Salesforce à l’aide d’un flux implicite OAuth pour authentifier l’utilisateur (déprécié)**

Dans ce scénario, la méthode **OAuth2** (Open Authorization 2.0) est adoptée pour autoriser **tSalesforceConnection** à accéder aux ressources de Salesforce. Ainsi, la sécurité est optimisée et les
utilisateurs n’ont pas besoin de saisir leur identifiant et mot de passe Salesforce directement dans le tSalesforceConnection, nécessaire pour le type d’authentification Basic.

Configurer le Job

Procédure
1. Déposez un tSalesforceConnection, un tFixedFlowInput et un tSalesforceOutput de la Palette dans l’espace de modélisation graphique.
2. Reliez le tFixedFlowInput au tSalesforceOutput à l’aide d’un lien Row > Main.

Configurer les composants

Procédure
1. Double-cliquez sur le tSalesforceConnection pour ouvrir sa vue Basic settings.

![Basic settings](image)

2. Dans la liste Connection Type, sélectionnez OAuth et, dans la liste Oauth2 flow type, sélectionnez Implicit Flow.
3. Dans les champs Client Id et Client Secret, saisissez respectivement la clé et la phrase secrète utilisés par le consommateur pour accéder à Salesforce.
4. Dans les champs Callback Host et Callback Port, saisissez respectivement l’hôte de rappel et le numéro de port.
5. Dans le champ Token File, parcourez votre système ou saisissez le chemin d’accès au fichier de jeton stockant le jeton de rafraîchissement utilisé par l’application connectée compatible OAuth pour obtenir de nouvelles sessions sans demander à l’utilisateur de fournir ses informations d’authentification.
7. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans l'éditeur de schéma, définissez-le en ajoutant une colonne **Name** de type **String**. Cela fait, cliquez sur **OK** pour sauvegarder les modifications et fermer la boîte de dialogue.

8. Dans la zone **Mode**, sélectionnez **Use Inline Content (delimited file)**. Dans le champ **Content** qui s'affiche, saisissez les données à écrire dans Salesforce, par exemple **Talend**.

9. Double-cliquez sur le **tSalesforceOutput** pour ouvrir sa vue **Basic settings**.

10. Dans la liste déroulante **Connection Component**, sélectionnez le composant **tSalesforceConnection** afin de réutiliser la connexion créée.

11. Cliquez sur le bouton [...] près du champ **Module Name** et, dans la boîte de dialogue, sélectionnez l'objet auquel vous souhaitez accéder. Dans cet exemple, sélectionnez **Account**.

12. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l'éditeur de schéma. Dans le panneau de droite, supprimez toutes les colonnes, sauf **Name**. Cela fait, cliquez sur **OK** pour sauvegarder les modifications et fermer la boîte de dialogue.
Exécuter le Job

Procédure

1. Appuyez sur les touches Ctrl + S pour sauvegarder le Job.
3. Copiez l’URL dans la barre d’adresse de votre navigateur. La page de connexion à Salesforce.com s’ouvre.
4. Saisissez votre identifiant et votre mot de passe puis cliquez sur le bouton **Log in to Salesforce**.
La réponse à l’authentification s’affiche dans le navigateur.

5. **Le Job continue à s’exécuter.**

Comme vous pouvez le constater dans la console, le Job est correctement exécuté.

**Scénario associé**

Effectuer un upsert sur des données Salesforce en se basant sur des ID externes à la page 3625
tSalesforceGetDeleted

Ce composant récupère les données supprimées d’un objet Salesforce sur une période spécifiée.

Propriétés du tSalesforceGetDeleted Standard

Ces propriétés sont utilisées pour configurer le tSalesforceGetDeleted s’exécutant dans le framework de Jobs Standard.

Le composant tSalesforceGetDeleted Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste <strong>Connection Component</strong>.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant. |

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Sélectionnez dans la liste déroulante le type de connexion. Les propriétés de la connexion varient selon le type de connexion sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Basic</strong> : sélectionnez cette option pour accéder à Salesforce en saisissant votre identifiant et votre mot de passe.</td>
</tr>
<tr>
<td></td>
<td>• <strong>OAuth</strong> : sélectionnez cette option pour accéder à Salesforce via le protocole OAuth (Open Authorization) 2.0 pour authentifier l’utilisateur. Dans la liste déroulante <strong>OAuth2 flow type</strong> qui s’affiche, sélectionnez un type de flux d’authentification OAuth 2.0, <strong>Json Web Token Flow</strong> ou <strong>Implicit Flow (Deprecated)</strong> (User Agent Flow). Ainsi, vous devez créer une application connectée dans Salesforce, afin de configurer un fournisseur d’authentification Salesforce.</td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant le flux d’authentification OAuth, consultez [Authenticate Apps with OAuth](en anglais).

Pour plus d’informations concernant la création d’une application connectée dans Salesforce, consultez [Create a Connected App](en anglais).

<table>
<thead>
<tr>
<th>User Id</th>
<th>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Mot de passe Salesforce associé à l’identifiant. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</td>
</tr>
<tr>
<td>Security Token</td>
<td>Jeton de sécurité Salesforce. Pour plus d’informations, consultez [Reset Your Security Token](en anglais). Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</td>
</tr>
<tr>
<td>Issuer</td>
<td>Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez [Create a Connected App](en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>Subject</td>
<td>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>Expiration time (in seconds)</td>
<td>Délai d’expiration de l’assertion (en secondes) durant les cinq prochains minutes. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>Key store password</td>
<td>Mot de passe du Keystore. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
</tbody>
</table>
| Client Id | Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez Create a Connected App (en anglais).

Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
|-----------------|--------------------------------------------------------------------------------------------------|

Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| Callback Host   | Valeur de l’hôte dans l’URL de la fonction de rappel pour l’authentification OAuth définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce.

Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| Callback Port   | Valeur du port dans l’URL de la fonction de rappel pour l’authentification OAuth, définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce.

Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| Token File      | Chemin d’accès au fichier de jeton stockant le jeton de rafraîchissement utilisé pour obtenir le jeton d’accès sans autorisation.

Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| Module Name     | Cliquez sur le bouton […] à côté du champ et, dans la boîte de dialogue qui s’ouvre, sélectionnez le module à utiliser, ou cochez la case Use custom object et spécifiez le nom du module dans le champ Object Name. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

• View schema : sélectionnez cette option afin de voir le schéma. |
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

### Start Date
Cliquez sur le bouton [...] à côté du champ et, dans le calendrier qui s'affiche, sélectionnez la date et l'heure à partir desquelles vous souhaitez récupérer les données.

Vous ne pouvez effectuer la recherche que sur les 30 derniers jours.

### End Date
Cliquez sur le bouton [...] à côté du champ et, dans le calendrier qui s'affiche, sélectionnez la date et l'heure avant lesquelles vous souhaitez récupérer les données.

### Advanced settings

<table>
<thead>
<tr>
<th>Salesforce URL</th>
<th>URL de Service Web, requise pour vous connecter à Salesforce.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>API version</strong></td>
<td>Version de l'API Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion <strong>OAuth</strong> est sélectionné.</td>
</tr>
<tr>
<td><strong>Need compression</strong></td>
<td>Cochez cette case pour activer la compression des messages SOAP, ce qui peut conduire à améliorer les performances.</td>
</tr>
<tr>
<td><strong>Use Http Chunked</strong></td>
<td>Cochez cette case pour utiliser le mécanisme de transfert des données HTTP morcelées.</td>
</tr>
<tr>
<td><strong>Client Id</strong></td>
<td>Saisissez l'ID d'un utilisateur réel afin de différencier les utilisateurs d'un même compte et mot de passe pour accéder au site Web de Salesforce.</td>
</tr>
<tr>
<td><strong>Timeout</strong></td>
<td>Saisissez, en millisecondes, le délai avant suspension des requêtes dans Salesforce.</td>
</tr>
<tr>
<td><strong>Use Proxy</strong></td>
<td>Cochez cette case pour utiliser un serveur proxy et, dans les champs <strong>Host</strong>, <strong>Port</strong>, <strong>User Id</strong> et <strong>Password</strong> qui s'affichent, spécifiez les paramètres de connexion du serveur proxy.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.</td>
</tr>
</tbody>
</table>
Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Utilisation

Règle d’utilisation

Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie.

Récupérer les données supprimées de Salesforce

Ce scénario décrit un Job à deux composants permettant de récupérer les données de compte supprimées de Salesforce, au cours des cinq derniers jours.

Configurer le Job pour récupérer des données supprimées de Salesforce

Procédure

1. Créez un nouveau Job, ajoutez un composant tSalesforceGetDeleted et un tLogRow, soit en saisissant leur nom dans l’espace de modélisation graphique, soit en les glissant depuis la Palette.
2. Reliez le tSalesforceGetDeleted au tLogRow à l’aide d’un lien de type Row > Main.

Configurer les composants pour récupérer des données supprimées de Salesforce

Procédure

1. Double-cliquez sur le composant tSalesforceGetDeleted pour afficher sa vue Basic settings.
2. Dans les champs **User Id**, **Password** et **Security Key**, saisissez les informations d’authentification de l’utilisateur pour accéder à Salesforce.

3. Cliquez sur le bouton [...] à côté du champ **Module Name** et, dans la boîte de dialogue qui s’ouvre, sélectionnez l’objet auquel vous souhaitez accéder. Dans ce scénario, sélectionnez l’objet **Account**. Cela fait, cliquez sur **OK** pour sauvegarder les paramètres et fermer la boîte de dialogue.

4. Cliquez sur le bouton [...] à côté du champ **Start Date** et, dans le calendrier qui s’affiche, sélectionnez la date et l’heure de début, afin de collecter uniquement les données supprimées depuis le moment spécifié. Dans cet exemple, sélectionnez **2016-06-08 00:00:00**.
5. Répétez l’opération pour la date et l’heure de fin dans le champ **End Date**, afin de collecter uniquement les données supprimées avant le moment spécifié. Dans cet exemple, sélectionnez 2016-06-13 00:00:00.

6. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.

7. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant précédent.

8. Dans la zone **Mode**, sélectionnez **Vertical (each row is a key/value list)** pour afficher les résultats sous forme de tableau dans la console.

**Exécuter le Job**

**Procédure**

1. Enregistrez votre Job en appuyant sur les touches **Ctrl+S**.
2. Appuyez sur **F6** pour exécuter le Job.
Comme affiché ci-dessus, les données supprimées durant les cinq derniers jours sont collectées et affichées sous forme de tableau dans la console.
**tSalesforceGetServerTimestamp**

Ce composant récupère la date actuelle sous forme de timestamp du serveur Salesforce.

### Propriétés du tSalesforceGetServerTimestamp Standard

Ces propriétés sont utilisées pour configurer le tSalesforceGetServerTimestamp s'exécutant dans le framework de Jobs Standard.

Le composant tSalesforceGetServerTimestamp Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

#### Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Type</strong></td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td>- <strong>Built-In</strong></td>
<td>les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>- <strong>Repository</strong></td>
<td>les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

| Connection Component | Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant. |

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Sélectionnez dans la liste déroulante le type de connexion. Les propriétés de la connexion varient selon le type de connexion sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td>- <strong>Basic</strong></td>
<td>sélectionnez cette option pour accéder à Salesforce en saisissant votre identifiant et votre mot de passe.</td>
</tr>
<tr>
<td>- <strong>OAuth</strong></td>
<td>sélectionnez cette option pour accéder à Salesforce via le protocole OAuth (Open Authorization) 2.0 pour authentifier l’utilisateur. Dans la liste déroulante OAuth2 flow type qui s’affiche, sélectionnez un type de flux d’authentification OAuth 2.0, Json Web Token Flow ou Implicit Flow (Deprecated) (User Agent Flow). Ainsi, vous devez créer une application connectée dans Salesforce, afin de configurer un fournisseur d’authentification Salesforce.</td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant le flux d’authentification OAuth, consultez *Authenticate Apps with OAuth* (en anglais).

Pour plus d’informations concernant la création d’une application connectée dans Salesforce, consultez *Create a Connected App* (en anglais).

<table>
<thead>
<tr>
<th>User Id</th>
<th>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion <strong>Basic</strong> est sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Mot de passe Salesforce associé à l’identifiant. Cette propriété est disponible uniquement lorsque le type de connexion <strong>Basic</strong> est sélectionné.</td>
</tr>
<tr>
<td>Security Token</td>
<td>Jeton de sécurité Salesforce. Pour plus d’informations, consultez <em>Reset Your Security Token</em> (en anglais). Cette propriété est disponible uniquement lorsque le type de connexion <strong>Basic</strong> est sélectionné.</td>
</tr>
<tr>
<td>Issuer</td>
<td>Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez <em>Create a Connected App</em> (en anglais). Cette propriété est disponible uniquement lorsque le type OAuth <strong>Json Web Token Flow</strong> est sélectionné.</td>
</tr>
<tr>
<td>Subject</td>
<td>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth <strong>Json Web Token Flow</strong> est sélectionné.</td>
</tr>
<tr>
<td>Expiration time (in seconds)</td>
<td>Délai d’expiration de l’assertion (en secondes) durant les cinq prochains minutes. Cette propriété est disponible uniquement lorsque le type OAuth <strong>Json Web Token Flow</strong> est sélectionné.</td>
</tr>
<tr>
<td>Key store password</td>
<td>Mot de passe du Keystore. Cette propriété est disponible uniquement lorsque le type OAuth <strong>Json Web Token Flow</strong> est sélectionné.</td>
</tr>
<tr>
<td><strong>Client Id</strong></td>
<td>Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez [Create a Connected App](en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Client Secret</strong></td>
<td>Consumer Secret d’OAuth, généré lorsque votre application connectée est créée et affichée dans la page des informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez [Create a Connected App](en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Callback Host</strong></td>
<td>Valeur de l’hôte dans l’URL de la fonction de rappel pour l’authentification OAuth définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Callback Port</strong></td>
<td>Valeur du port dans l’URL de la fonction de rappel pour l’authentification OAuth, définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Token File</strong></td>
<td>Chemin d’accès au fichier de jeton stockant le jeton de rafraîchissement utilisé pour obtenir le jeton d’accès sans autorisation. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
</tbody>
</table>
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• **View schema** : sélectionnez cette option afin de voir le schéma.
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les |
modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Advanced settings**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salesforce URL</td>
<td>URL de Service Web, requise pour vous connecter à Salesforce.</td>
</tr>
<tr>
<td>API version</td>
<td>Version de l'API Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion OAuth est sélectionné.</td>
</tr>
<tr>
<td>Use or save the connection session</td>
<td>Cochez cette case et, dans le champ Session directory qui s’affiche, spécifiez le chemin vers le fichier de session de connexion à sauvegarder ou utiliser. Ce fichier de session peut être partagé par différents Jobs afin de récupérer une session de connexion, tant que le bon ID utilisateur est fourni par le composant. Ainsi, vous n'avez pas besoin de vous connecter au serveur pour récupérer la session. Lorsqu'une session expirée est détectée, si les informations de connexion sont fournies (ID utilisateur, mot de passe et clé de sécurité) le composant va se connecter au serveur pour récupérer les informations de la nouvelle session et mettre à jour le fichier de session de connexion. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</td>
</tr>
<tr>
<td>Need compression</td>
<td>Cochez cette case pour activer la compression des messages SOAP, ce qui peut conduire à améliorer les performances.</td>
</tr>
<tr>
<td>Use Http Chunked</td>
<td>Cochez cette case pour utiliser le mécanisme de transfert des données HTTP morcelées.</td>
</tr>
<tr>
<td>Client Id</td>
<td>Saisissez l’ID d’un utilisateur réel afin de différencier les utilisateurs d’un même compte et mot de passe pour accéder au site Web de Salesforce.</td>
</tr>
<tr>
<td>Timeout</td>
<td>Saisissez, en millisecondes, le délai avant suspension des requêtes dans Salesforce.</td>
</tr>
<tr>
<td>Use Proxy</td>
<td>Cochez cette case pour utiliser un serveur proxy et, dans les champs Host, Port, User Id et Password qui s’affiche, spécifiez les paramètres de connexion du serveur proxy.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>
**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie. |

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tSalesforceGetUpdated

Ce composant récupère les données mises à jour d’un objet Salesforce sur une période spécifiée.

Propriétés du tSalesforceGetUpdated Standard

Ces propriétés sont utilisées pour configurer le tSalesforceGetUpdated s’exécutant dans le framework de Jobs Standard.

Le composant tSalesforceGetUpdated Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.

| Connection Component | Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant. |

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Sélectionnez dans la liste déroulante le type de connexion. Les propriétés de la connexion varient selon le type de connexion sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Basic</strong> : sélectionnez cette option pour accéder à Salesforce en saisissant votre identifiant et votre mot de passe.</td>
</tr>
<tr>
<td></td>
<td>• <strong>OAuth</strong> : sélectionnez cette option pour accéder à Salesforce via le protocole OAuth (Open Authorization) 2.0 pour authentifier l’utilisateur. Dans la liste déroulante OAuth2 flow type qui s’affiche, sélectionnez un type de flux d’authentification OAuth 2.0, Json Web Token Flow ou Implicit Flow (Deprecated) (User Agent Flow). Ainsi, vous devez créer une application connectée dans Salesforce, afin de configurer un fournisseur d’authentification Salesforce.</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant la création d’une application connectée dans Salesforce, consultez [Create a Connected App](https://docs.salesforce.com/en_us/developer/docs/), (en anglais).

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Id</td>
<td>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</td>
</tr>
<tr>
<td>Password</td>
<td>Mot de passe Salesforce associé à l’identifiant. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</td>
</tr>
<tr>
<td>Subject</td>
<td>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>Expiration time (in seconds)</td>
<td>Délai d’expiration de l’assertion (en secondes) durant les cinq prochains minutes. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>Key store password</td>
<td>Mot de passe du Keystore. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Client Id</strong></td>
<td>Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez Create a Connected App (en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Client Secret</strong></td>
<td>Consumer Secret d’OAuth, généré lorsque votre application connectée est créée et affichée dans la page des informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez Create a Connected App (en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Callback Host</strong></td>
<td>Valeur de l’hôte dans l’URL de la fonction de rappel pour l’authentification OAuth définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Callback Port</strong></td>
<td>Valeur du port dans l’URL de la fonction de rappel pour l’authentification OAuth, définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Token File</strong></td>
<td>Chemin d’accès au fichier de jeton stockant le jeton de rafraîchissement utilisé pour obtenir le jeton d’accès sans autorisation. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Module Name</strong></td>
<td>Cliquez sur le bouton [...] à côté du champ et, dans la boîte de dialogue qui s’ouvre, sélectionnez le module à utiliser, ou cochez la case Use custom object et spécifiez le nom du module dans le champ Object Name.</td>
</tr>
</tbody>
</table>
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
  - View schema : sélectionnez cette option afin de voir le schéma. |
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

| **Start Date** | Cliquez sur le bouton [...] à côté du champ et, dans le calendrier qui s’affiche, sélectionnez la date et l’heure à partir desquelles vous souhaitez récupérer les données. Vous ne pouvez effectuer la recherche que sur les 30 derniers jours. |
| **End Date** | Cliquez sur le bouton [...] à côté du champ et, dans le calendrier qui s’affiche, sélectionnez la date et l’heure avant lesquelles vous souhaitez récupérer les données. |

**Advanced settings**

| **Salesforce URL** | URL de Service Web, requise pour vous connecter à Salesforce. |
| **API version** | Version de l’API Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion **OAuth** est sélectionné. |
| **Use or save the connection session** | Cochez cette case et, dans le champ **Session directory** qui s’affiche, spécifiez le chemin vers le fichier de session de connexion à sauvegarder ou utiliser.

Ce fichier de session peut être partagé par différents Jobs afin de récupérer une session de connexion, tant que le bon ID utilisateur est fourni par le composant. Ainsi, vous n’avez pas besoin de vous connecter au serveur pour récupérer la session.

Lorsqu’une session expirée est détectée, si les informations de connexion sont fournies (ID utilisateur, mot de passe et clé de sécurité) le composant va se connecter au serveur pour récupérer les informations de la nouvelle session et mettre à jour le fichier de session de connexion.

Cette propriété est disponible uniquement lorsque le type de connexion **Basic** est sélectionné. |
<p>| <strong>Need compression</strong> | Cochez cette case pour activer la compression des messages SOAP, ce qui peut conduire à améliorer les performances. |</p>
<table>
<thead>
<tr>
<th><strong>Use Http Chunked</strong></th>
<th>Cochez cette case pour utiliser le mécanisme de transfert des données HTTP morcelées.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Client Id</strong></td>
<td>Saisissez l'ID d'un utilisateur réel afin de différencier les utilisateurs d'un même compte et mot de passe pour accéder au site Web de Salesforce.</td>
</tr>
<tr>
<td><strong>Timeout</strong></td>
<td>Saisissez, en millisecondes, le délai avant suspension des requêtes dans Salesforce.</td>
</tr>
<tr>
<td><strong>Use Proxy</strong></td>
<td>Cochez cette case pour utiliser un serveur proxy et, dans les champs <code>Host</code>, <code>Port</code>, <code>User Id</code> et <code>Password</code> qui s’affiche, spécifiez les paramètres de connexion du serveur proxy.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th><strong>NB_LINE</strong></th>
<th>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th><strong>Règle d'utilisation</strong></th>
<th>Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie.</th>
</tr>
</thead>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tSalesforceInput

Ce composant récupère les données d’un objet Salesforce à l’aide d’une requête.

**Propriétés du tSalesforceInput Standard**

Ces propriétés sont utilisées pour configurer le tSalesforceInput s’exécutant dans le framework de Jobs Standard.

Le composant tSalesforceInput Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Type</strong></td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td>• <strong>Built-In</strong></td>
<td>les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>• <strong>Repository</strong></td>
<td>les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connection Component</th>
<th>Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sélectionnez dans la liste déroulante le type de connexion. Les propriétés de la connexion varient selon le type de connexion sélectionné.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Basic</strong></td>
<td>sélectionnez cette option pour accéder à Salesforce en saisissant votre identifiant et votre mot de passe.</td>
</tr>
<tr>
<td>• <strong>OAuth</strong></td>
<td>sélectionnez cette option pour accéder à Salesforce via le protocole OAuth (Open Authorization) 2.0 pour authentifier l’utilisateur. Dans la liste déroulante OAuth2 flow type qui s’affiche, sélectionnez un type de flux d’authentification OAuth 2.0, Json Web Token Flow ou Implicit Flow (Deprecated) (User Agent Flow). Ainsi, vous devez créer une application connectée dans Salesforce, afin de configurer un fournisseur d’authentification Salesforce.</td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant le flux d’authentification OAuth, consultez [Authenticate Apps with OAuth](https://docs.salesforce.com/Documentation/oauthguide) (en anglais).

Pour plus d’informations concernant la création d’une application connectée dans Salesforce, consultez [Create a Connected App](https://docs.salesforce.com/Documentation/appconnect/appstream) (en anglais).

<table>
<thead>
<tr>
<th>User Id</th>
<th>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Mot de passe Salesforce associé à l’identifiant. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</td>
</tr>
<tr>
<td>Subject</td>
<td>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>Expiration time (in seconds)</td>
<td>Durée d’expiration de l’assertion (en secondes) durant les cinq prochains minutes. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>Key store password</td>
<td>Mot de passe du Keystore. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
</tbody>
</table>
| **Client Id** | Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez Create a Connected App (en anglais).

Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| **Client Secret** | Consumer Secret d’OAuth, généré lorsque votre application connectée est créée et affichée dans la page des informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez Create a Connected App (en anglais).

Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| **Callback Host** | Valeur de l’hôte dans l’URL de la fonction de rappel pour l’authentification OAuth définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce.

Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| **Callback Port** | Valeur du port dans l’URL de la fonction de rappel pour l’authentification OAuth, définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce.

Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| **Token File** | Chemin d’accès au fichier de jeton stockant le jeton de rafraîchissement utilisé pour obtenir le jeton d’accès sans autorisation.

Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| **Module Name** | Cliquez sur le bouton [...] à côté du champ et, dans la boîte de dialogue qui s’ouvre, sélectionnez le module à utiliser, ou cochez la case Use custom object et spécifiez le nom du module dans le champ Object Name. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma. |
### Change to built-in property

Sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

### Update repository connection

Sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Dans ce composant, le schéma est relatif au Module sélectionné.

Pour retrouver une colonne d’un module lié, il est nécessaire de définir la colonne, d’une manière précise, dans la vue Edit Schema, sinon la requête relationnelle ne fonctionnera pas. La syntaxe correcte est : NameofCurrentModule_NameofLinkedModule_NameofColumnofInterest

### Query Mode

Sélectionnez dans la liste déroulante le mode de requête, Query ou Bulk.

- **Query** : une requête simple, ordinaire.
- **Bulk** : une requête de masse utilisée pour interroger des jeux de données volumineux de manière efficace et réduire le nombre de requêtes de l’API.

### Condition

Saisissez entre guillemets doubles la requête utilisée pour sélectionner les données à extraire. Par exemple : "name='Talend'" ou "name like '%talend_user%'".

Cette propriété est disponible uniquement lorsque la case Manual Query est cochée.

### Manual query

Cochez cette case et, dans le champ Full SOQL query string qui s’affiche, saisissez, entre guillemets doubles, l’instruction complète SOQL (Salesforce Object Query Language) utilisée pour sélectionner les données à récupérer. Pour plus d’informations concernant SOQL, consultez Salesforce Object Query Language (SOQL) (en anglais).

### Guess schema

Cliquez sur ce bouton pour générer les colonnes du schéma à l’aide d’une requête spécifiée dans le champ Full SOQL query string.

Cette propriété est disponible uniquement lorsque la case Manual Query est cochée.

Notez que certaines limitations existent. Vous devez les prendre en compte lorsque vous utilisez cette fonctionnalité. Veuillez vous référer au chapitre Know Limitations dans les Release Notes (en anglais) pour plus de détails.

### Guess query

Cliquez sur ce bouton pour générer la requête dans le champ Full SOQL query string, en se basant sur le module et le schéma définis.
Pour plus d’informations relatives à la bonne configuration d’un module ou d’un schéma lors de la génération de requêtes SOQL, consultez Configurer le schéma pour la fonctionnalité de récupération de requêtes du tSalesforceInput à la page 3613.

Cette propriété est disponible uniquement lorsque la case Manual Query est cochée.

Notez que certaines limitations existent. Vous devez les prendre en compte lorsque vous utilisez cette fonctionnalité. Veuillez vous référer au chapitre Know Limitations dans les Release Notes (en anglais) pour plus de détails.

### Advanced settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include deleted records</td>
<td>Cochez cette case pour interroger tous les enregistrements, y compris les enregistrements supprimés. Cette propriété est disponible uniquement lorsque l’option Query est sélectionnée dans la liste déroulante Query Mode.</td>
</tr>
<tr>
<td><strong>Salesforce URL</strong></td>
<td>URL de Service Web, requise pour vous connecter à Salesforce.</td>
</tr>
<tr>
<td><strong>API version</strong></td>
<td>Version de l’API Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion OAuth est sélectionné.</td>
</tr>
<tr>
<td><strong>Use or save the connection session</strong></td>
<td>Cochez cette case et, dans le champ Session directory qui s’affiche, spécifiez le chemin vers le fichier de session de connexion à sauvegarder ou utiliser. Ce fichier de session peut être partagé par différents Jobs afin de récupérer une session de connexion, tant que le bon ID utilisateur est fourni par le composant. Ainsi, vous n’avez pas besoin de vous connecter au serveur pour récupérer la session. Lorsqu’une session expirée est détectée, si les informations de connexion sont fournies (ID utilisateur, mot de passe et clé de sécurité) le composant va se connecter au serveur pour récupérer les informations de la nouvelle session et mettre à jour le fichier de session de connexion. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</td>
</tr>
<tr>
<td><strong>Need compression</strong></td>
<td>Cochez cette case pour activer la compression des messages SOAP, ce qui peut conduire à améliorer les performances.</td>
</tr>
<tr>
<td><strong>Trace HTTP message</strong></td>
<td>Cochez cette case pour écrire en sortie les interactions HTTP dans la console. Cette propriété est disponible uniquement lorsque l’option Bulk est sélectionnée dans la liste Query Mode.</td>
</tr>
<tr>
<td>Use HTTP Chunked</td>
<td>Cochez cette case pour utiliser le mécanisme de transfert des données HTTP morcelées. Cette propriété est disponible uniquement lorsque l’option Query est sélectionnée dans la liste déroulante Query Mode.</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Client Id</td>
<td>Saisissez l’ID d’un utilisateur réel afin de différencier les utilisateurs d’un même compte et mot de passe pour accéder au site Web de Salesforce.</td>
</tr>
<tr>
<td>Timeout</td>
<td>Saisissez, en millisecondes, le délai avant suspension des requêtes dans Salesforce.</td>
</tr>
<tr>
<td>Use Proxy</td>
<td>Cochez cette case pour utiliser un serveur proxy et, dans les champs Host, Port, User Id et Password qui s’affiche, spécifiez les paramètres de connexion du serveur proxy.</td>
</tr>
<tr>
<td>Batch Size</td>
<td>Saisissez le nombre d’enregistrements dans chaque lot traité. Cette propriété est disponible uniquement lorsque l’option Query est sélectionnée dans la liste déroulante Query Mode.</td>
</tr>
<tr>
<td>Normalize Delimiter</td>
<td>Saisissez le caractère, la chaîne ou l’expression régulière utilisé(e) pour normaliser les données collectées à l’aide de requêtes effectuées sur les relations hiérarchiques entre les différents objets Salesforce. Cette propriété est disponible uniquement lorsque l’option Query est sélectionnée dans la liste déroulante Query Mode.</td>
</tr>
<tr>
<td>Column Name Delimiter</td>
<td>Saisissez le caractère, la chaîne ou l’expression régulière utilisé pour séparer le nom de l’objet parent du nom de l’objet fils lorsque vous effectuez une requête sur les relations hiérarchiques entre les différents objets Salesforce. Cette propriété est disponible uniquement lorsque l’option Query est sélectionnée dans la liste déroulante Query Mode.</td>
</tr>
<tr>
<td>Safety Switch</td>
<td>Décochez cette case si la longueur d’une colonne du module à interroger est supérieure à 100 000 caractères. Par défaut, cette case est cochée pour éviter une consommation excessive de mémoire. Cette propriété est disponible uniquement lorsque l’option Bulk est sélectionnée dans la liste Query Mode.</td>
</tr>
<tr>
<td>Complete Job timeout</td>
<td>Valeur du délai avant suspension, en secondes, au cours de laquelle votre Job de requête de masse doit être terminé. Par défaut, la valeur est configurée à 0, ce qui signifie qu’il n’y a aucune limite de temps pour la fin de votre Job. Salesforce a sa propre limite pour les requêtes de masse. Si le traitement d’un lot dépasse la limite de Salesforce, le lot est placé dans une file et traité ultérieurement. Le Job peut donc prendre trop de temps pour s’exécuter.</td>
</tr>
</tbody>
</table>
Dans ce cas, vous pouvez définir la valeur du délai avant suspension dans ce champ pour terminer le Job en avance. Pour plus d’informations concernant la limite de Salesforce, consultez Bulk API Limits - Batch processing time (en anglais).

Cette propriété est disponible uniquement lorsque l’option Bulk est sélectionnée dans la liste Query Mode.

**Enable PK Chunking**

Cochez cette case pour autoriser l’encodage par morceau de la PK (clé primaire, c’est-à-dire, l’identifiant de l’enregistrement de l’objet) lors de l’extraction de grands volumes d’enregistrements ou si la requête est systématiquement suspendue. Dans le champ Chunk size affiché, spécifiez le nombre d’enregistrements dans les limites de l’identifiant pour chaque morceau. Pour plus d’informations, consultez Use PK Chunking to Extract Large Data Sets from Salesforce (en anglais).

Cette propriété est disponible uniquement lorsque l’option Bulk est sélectionnée dans la liste Query Mode.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Global Variables**

<table>
<thead>
<tr>
<th>NB_LINE</th>
<th>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie. |

**Configurer le schéma pour la fonctionnalité de récupération de requêtes du tSalesforceInput**

Le tSalesforceInput vous permet de générer des requêtes SOQL, en se basant sur le module et le schéma définis, à la fois sur des champs et des éléments standards et personnalisés. Cette section vous explique comment définir le module et le schéma lors de la génération d’une requête SOQL.

Il existe deux types de requêtes SOQL : la requête simple et la requête de relation.

La requête simple récupère uniquement les données d’un élément. Pour plus d’informations relatives à la bonne configuration du module et du schéma pour générer des requêtes simples, consultez Configuration du schéma pour générer la requête simple à la page 3614.
La requête de relation récupère des données à partir de plusieurs types d’élément, y compris les requêtes de relation enfant-parent et parent-enfant.

Pour plus d’informations relatives à la bonne configuration du module et du schéma pour générer des requêtes de relation enfant-parent, consultez Configuration du schéma pour générer la requête de relation enfant-parent à la page 3614.

Pour plus d’informations relatives à la bonne configuration du module et du schéma pour générer des requêtes de relation, consultez Configuration du schéma pour générer la requête de relation parent-enfant à la page 3616.

Avant de configurer le nom du module et les colonnes du schéma pour générer la requête de relation dans la vue Basic settings d’un tSalesforceInput, vous devez identifier le type de relation (enfant-parent ou parent-enfant) de la requête à générer, car les méthodes de configuration des noms de colonnes du schéma sont différentes pour les deux types de requêtes de relations. La différence principale entre ces deux types de requêtes est que la requête parent-enfant est spécifiée par l’utilisation de la sous-requête contenue dans les parenthèses, alors que la requête enfant-parent ne l’est pas. Pour plus d’informations concernant les requêtes de relations en SOQL et comment identifier les requêtes de relations, consultez Relationship Queries (en anglais).

Configuration du schéma pour générer la requête simple

Cette section vous explique comment définir le nom du module et les colonnes du schéma pour générer la requête simple.

Les deux requêtes simples ci-dessous seront utilisées à des fins de démonstration dans les étapes suivantes :

- SELECT Id, Name, BillingCity FROM Account, une requête simple avec un élément et des champs standards et
- SELECT Name__c, LastName__c FROM Mother__c, une requête simple avec un élément et des champs personnalisés.

Procédure

1. Définissez le nom du module avec le nom de l’élément spécifié dans la clause FROM, Account et Mother__c pour les exemples ci-dessus.
2. Créez une colonne pour chaque champ dans la liste des champs (séparés par une virgule) après SELECT dans la boîte de dialogue du schéma et définissez le nom du champ comme le nom de la colonne.
   
   Pour le premier exemple, vous devez créer trois colonnes Id, Name et BillingCity pour les trois champs.
   
   Pour le second exemple, vous devez créer deux colonnes Name__c et LastName__c pour les deux champs.

Configuration du schéma pour générer la requête de relation enfant-parent

Cette section explique comment configurer le nom du module et les colonnes du schéma pour générer une requête de relation enfant-parent.

Les deux exemples de requête relation enfant-parent suivants seront utilisés à des fins de démonstration dans les étapes suivantes :
• SELECT Name, Account.Name, Account.Owner.Name FROM Contact, une requête de relation parent-enfant avec un élément et des champs standards, et
• SELECT Id, FirstName__c, MotherOfDaughter__r.FirstName__c FROM Daughter__c, une requête de relation enfant-parent avec un élément et des champs personnalisés.

Notez que vous devez utiliser ici un nom de relation avec __r au lieu de __c. Pour plus d'informations, consultez Understanding Relationship Names, Custom Objects, and Custom Fields (en anglais).

Procédure

1. Configurez le nom du module avec le nom de l'objet spécifié dans la clause FROM, Contact et Daughter__c dans les exemples ci-dessus.
2. Créez une colonne pour chaque champ dans la liste des champs (séparés par une virgule) après SELECT dans la boîte de dialogue du schéma.
   
   Pour le premier exemple, vous devez créer trois colonnes pour les trois champs Name, Account.Name et Account.Owner.Name.
   
   Pour le second exemple, vous devez créer trois colonnes pour les trois champs Id, FirstName__c et MotherOfDaughter__r.FirstName__c.
3. Configurez le nom de chaque colonne avec le nom de chaque champ et remplacez tous les points du nom de la colonne par des tirets bas.
   
   Pour le premier exemple, les noms des trois colonnes sont définis comme Name, Account_Name et Account__Owner_Name.
   
   Pour le second exemple, les noms des trois colonnes sont définis comme Id, FirstName__c et MotherOfDaughter__r_FirstName__c.
4. Configurez le type de chaque colonne.

Le schéma pour le premier exemple doit être configuré comme suit :

Le schéma pour le second exemple doit être configuré comme suit :
Le caractère tiret bas '_' est utilisé en tant que séparateur entre le nom de la relation et le nom du champ dans le schéma Talend. Seul le tiret bas '_' après '__r' ou '__c' sera remplacé par le caractère point '.' lors de la génération de la requête. Si le tiret bas '_' fait partie d'un nom personnalisé dans le schéma, par exemple, `Contact_custom_field__c` devrait être `Contact.custom_field__c` dans la requête, vous devez remplacer '_' dans la requête générée par '.', manuellement.

**Configuration du schéma pour générer la requête de relation parent-enfant**

Cette section explique comment configurer le nom du module et les colonnes du schéma pour générer une requête de relation parent-enfant.

Les deux exemples suivants de requête de relation parent-enfant seront utilisés à des fins de démonstration dans les étapes suivantes :

- `SELECT Name, Owner.Name (SELECT CreatedBy.Name FROM Notes) FROM Account`, une requête de relation parent-enfant avec un élément et des champs standards et
- `SELECT LastName__c, (SELECT FirstName__c FROM Daughters__r) FROM Mother__c`, une requête de relation parent-enfant avec un élément et des champs personnalisés.

Notez que vous devez utiliser ici un nom de relation avec __r au lieu de __c. Pour plus d'informations, consultez [Understanding Relationship Names, Custom Objects, and Custom Fields](en anglais).

**Procédure**

1. Configurez le nom du module avec le nom de l'objet spécifié dans la requête extérieure (clause FROM), `Account` et `Mother__c` dans les exemples ci-dessus.

2. Créez une colonne pour chaque champ (y compris les champs dans la sous-requête) après `SELECT` dans la fenêtre de schéma.

Pour le premier exemple, vous devez créer trois colonnes pour les trois champs, y compris les deux champs `Name` et `Owner.Name` après la clause extérieure `SELECT` et un champ `CreatedBy.Name` après la sous-requête `SELECT`.

Pour le second exemple, vous devez créer deux colonnes pour les deux champs, y compris le champ `LastName__c` après la clause extérieure `SELECT` et le champ `FirstName__c` après la sous-requête `SELECT`. 
3. Pour les champs dans la clause extérieure SELECT, qui sont hors des parenthèses, configurez le nom de chaque colonne avec le nom de chaque champ et remplacer tous les points du nom de colonne par des tirets bas.

Pour le premier exemple, les noms de colonnes des deux champs Name et Owner.Name dans la clause extérieure SELECT sont définis comme Name et Owner_Name.

Pour le second exemple, il n’y pas de “.” dans le nom du fichier, afin que le nom de la colonne soit le même que le nom du champ.

4. Pour les champs dans la sous-requête SELECT, construisez les noms de colonnes en utilisant le modèle <$XXX>_records_<$YYY>, où <$XXX> correspond au nom de l’objet spécifié dans la sous-requête (clause FROM) et <$YYY> sera le nom du champ, une fois les points remplacés par des tirets bas.

Pour le premier exemple, le nom de la colonne pour le champ CreatedBy.Name dans la sous-requête est défini comme Notes_records_CreatedBy_Name.

Pour le second exemple, le nom de la colonne pour le champ FirstName__c dans la sous-requête est défini à Daughters__r_records_FirstName__c.

5. Configurez le type de chaque colonne.

Le schéma pour le premier exemple doit être défini comme suit :

![Schema of tSalesforceInput_1](image1)

Le schéma pour le second exemple doit être défini comme suit :

![Schema of tSalesforceInput_1](image2)
Le caractère tiret bas '_' est utilisé en tant que séparateur entre le nom de la relation et le nom du champ dans le schéma Talend. Seul le tiret bas '_' après '__r' ou '__c' sera remplacé par le caractère point '.' lors de la génération de la requête. Si le tiret bas '_' fait partie d'un nom personnalisé dans le schéma, par exemple, Contact_custom_field__c, devrait être Contact.custom_field__c dans la requête, vous devez remplacer '_' dans la requête générée par '.', manuellement.

**Scénario associé**

Effectuer un upsert sur des données Salesforce en se basant sur des ID externes à la page 3625
**tSalesforceOutput**

Ce composant insère, met à jour, effectue un upsert ou supprime des données dans un objet Salesforce.

### Propriétés du tSalesforceOutput Standard

Ces propriétés sont utilisées pour configurer le tSalesforceOutput s’exécutant dans le framework de Jobs Standard.

Le composant tSalesforceOutput Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

#### Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-In</strong></td>
<td>les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.

| Connection Component | Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant. |

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Sélectionnez dans la liste déroulante le type de connexion. Les propriétés de la connexion varient selon le type de connexion sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic</strong></td>
<td>sélectionnez cette option pour accéder à Salesforce en saisissant votre identifiant et votre mot de passe.</td>
</tr>
<tr>
<td><strong>OAuth</strong></td>
<td>sélectionnez cette option pour accéder à Salesforce via le protocole OAuth (Open Authorization) 2.0 pour authentifier l’utilisateur. Dans la liste déroulante Oauth2 flow type qui s’affiche, sélectionnez un type de flux d’authentification OAuth 2.0, Json Web Token Flow ou Implicit Flow (Deprecated) (User Agent Flow). Ainsi, vous devez créer une application connectée dans Salesforce, afin de configurer un fournisseur d’authentification Salesforce.</td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant le flux d’authentification OAuth, consultez [Authenticate Apps with OAuth](en anglais).
Pour plus d’informations concernant la création d’une application connectée dans Salesforce, consultez [Create a Connected App](en anglais).

| **User Id** | Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné. |
| **Password** | Mot de passe Salesforce associé à l’identifiant. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné. |
| **Security Token** | Jeton de sécurité Salesforce. Pour plus d’informations, consultez [Reset Your Security Token](en anglais). Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné. |
| **Issuer** | Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez [Create a Connected App](en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné. |
| **Subject** | Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné. |
| **Expiration time (in seconds)** | Délai d’expiration de l’assertion (en secondes) durant les cinq prochains minutes. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné. |
| **Key store** | Chemin pointant vers le fichier Keystore au format Java Keystore (JKS). Le fichier Keystore peut être généré en créant un certificat signé par Salesforce et en l’exportant vers le Keystore. Pour plus d’informations, consultez [Generate a Self-Signed Certificate](en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné. |
| **Key store password** | Mot de passe du Keystore. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné. |
| **Certificate alias** | Nom unique du certificat signé par Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné. |
| **Client Id** | Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez Create a Connected App (en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| **Client Secret** | Consumer Secret d’OAuth, généré lorsque votre application connectée est créée et affichée dans la page des informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez Create a Connected App (en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| **Callback Host** | Valeur de l’hôte dans l’URL de la fonction de rappel pour l’authentification OAuth définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| **Callback Port** | Valeur du port dans l’URL de la fonction de rappel pour l’authentification OAuth, définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| **Token File** | Chemin d’accès au fichier de jeton stockant le jeton de rafraîchissement utilisé pour obtenir le jeton d’accès sans autorisation. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné. |
| **Module Name** | Cliquez sur le bouton [...] à côté du champ et, dans la boîte de dialogue qui s’ouvre, sélectionnez le module à utiliser, ou cochez la case Use custom object et spécifiez le nom du module dans le champ Object Name. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles : • View schema : sélectionnez cette option afin de voir le schéma. |
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

### Output Action

Sélectionnez dans la liste l’une des opérations suivantes à effectuer.

- **INSERT** : insérer un ou plusieurs enregistrements dans Salesforce.
- **UPDATE** : mettre à jour un ou plusieurs enregistrements existants dans Salesforce.
- **UPSERT** : créer de nouveaux enregistrements et mettre à jour des enregistrements existants. Dans le champ **Upsert Key Column** affiché, spécifiez la colonne clé pour l’opération d’upsert.
- **DELETE** : supprimer un ou plusieurs enregistrements dans Salesforce.

### Advanced settings

<table>
<thead>
<tr>
<th>Salesforce URL</th>
<th>URL de Service Web, requise pour vous connecter à Salesforce.</th>
</tr>
</thead>
<tbody>
<tr>
<td>API version</td>
<td>Version de l’API Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion <strong>OAuth</strong> est sélectionné.</td>
</tr>
<tr>
<td>Use or save the connection session</td>
<td>Cochez cette case et, dans le champ <strong>Session directory</strong> qui s’affiche, spécifiez le chemin vers le fichier de session de connexion à sauvegarder ou utiliser. Ce fichier de session peut être partagé par différents Jobs afin de récupérer une session de connexion, tant que le bon ID utilisateur est fourni par le composant. Ainsi, vous n’avez pas besoin de vous connecter au serveur pour récupérer la session. Lorsqu’une session expirée est détectée, si les informations de connexion sont fournies (ID utilisateur, mot de passe et clé de sécurité) le composant va se connecter au serveur pour récupérer les informations de la nouvelle session et mettre à jour le fichier de session de connexion. Cette propriété est disponible uniquement lorsque le type de connexion <strong>Basic</strong> est sélectionné.</td>
</tr>
</tbody>
</table>
### Need compression
Cochez cette case pour activer la compression des messages SOAP, ce qui peut conduire à améliorer les performances.

### Use Http Chunked
Cochez cette case pour utiliser le mécanisme de transfert des données HTTP morcelées.

### Client Id
Saisissez l’ID d’un utilisateur réel afin de différencier les utilisateurs d’un même compte et mot de passe pour accéder au site Web de Salesforce.

### Timeout
Saisissez, en millisecondes, le délai avant suspension des requêtes dans Salesforce.

### Use Proxy
Cochez cette case pour utiliser un serveur proxy et, dans les champs Host, Port, User Id et Password qui s’affiche, spécifiez les paramètres de connexion du serveur proxy.

### Relationship mapping for upsert
Cliquez sur le bouton [+] pour ajouter des lignes et spécifiez les champs d’ID externes du flux d’entrée, les champs de relations lookup dans le module d’upsert, le module de lookup ainsi que les champs d’ID externes dans le module de lookup.


- **Lookup field name** : nom du champ de lookup. Cela fait référence au champ de lookup du module spécifique dans le champ Module Name de la vue Basic settings. Cette colonne doit être spécifiée lorsque des valeurs d’entrée de cette colonne sont définies comme NULL et que la case Ignore Null est décochée.

- **Lookup relationship field name** : nom du champ de relations lookup. Cela fait référence aux champs de relations lookup spécifiés dans le champ Module Name de la vue Basic settings. Ils servent à établir les relations avec le module de lookup spécifié dans la colonne Module name de cette table. Pour plus d’informations concernant la définition des champs de relations lookup et comment saisir le bon nom dans le champ Lookup field name, rendez-vous sur le site Web de Salesforce et lancez l’application Salesforce Data Loader.

- **Module name** : nom du module de lookup.

- **External id name** : nom du champ d’ID externe dans le module de lookup spécifié dans la colonne Module name.

Cette propriété est disponible uniquement lorsque l’option UPSEMT est sélectionnée dans la liste déroulante Output Action.
### tSalesforceOutput

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extend Insert</strong></td>
<td>Cochez cette case pour transférer les données de sortie par lots. Dans le champ <strong>Commit Level</strong> qui s’affiche, spécifiez le nombre de lignes par lot.</td>
</tr>
<tr>
<td><strong>Cease on Error</strong></td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien <strong>Row &gt; Reject</strong>. La case <strong>Extend Insert</strong> doit être décochée lorsque vous utilisez le lien <strong>Row &gt; Reject</strong> afin de collecter les lignes en erreur.</td>
</tr>
</tbody>
</table>
| **Retrieve Id**               | Cochez cette case pour permettre à Salesforce de retourner l’ID Salesforce de l’enregistrement inséré ou mis à jour. Cette case est disponible uniquement lorsque l’option **INSERT** ou **UPSERT** est sélectionnée dans la liste **Output Action** et que la case **Extend Insert** est décochée. Lorsque cette case est cochée et que le composant **tSalesforceOutput** est relié à un autre composant via un lien **Row > Main**, les colonnes suivantes seront ajoutées au schéma du flux de données :  
  • **salesforce_id** : ID Salesforce de l’enregistrement créé ou mis à jour.  
  • **salesforce_upsert_status** (pour **UPSERT** uniquement) : statut de l’action d’upsert, indiquant si l’enregistrement est créé ou mis à jour. |
| **Ignore Null**               | Cochez cette case pour ignorer les valeurs NULL. Cette propriété est disponible uniquement lorsque l’option **UPDATE** ou **UPSERT** est sélectionnée dans la liste déroulante **Output Action**.                                                                                                                                                                                                                                                                                                                                                                                                          |
| **Log File Name**             | Spécifiez le chemin d’accès au fichier de log contenant tous les logs d’erreur.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| **tStatCatcher Statistics**   | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| **Global Variables**          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| **NB_LINE**                   | Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| **NB_SUCCESS**                | Nombre de lignes traitées correctement. Cette variable est une variable After et retourne un entier.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| **NB_REJECT**                 | Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

Utilisation

Règle d’utilisation | Ce composant peut être utilisé en tant que composant de fin ou composant intermédiaire dans un Job ou un sous-job et nécessite un lien d’entrée.

Effectuer un upsert sur des données Salesforce en se basant sur des ID externes

Voici un exemple d’utilisation des composants Talend pour insérer et mettre à jour des données d’un objet Salesforce Contact, en vous basant sur les relations de mapping avec les ID externes de l’objet Account.

Le Job utilisé dans cet exemple se compose des sept sous-jobs suivants :

- le premier sous-job ouvre une connexion à Salesforce,
- le deuxième supprime des enregistrements Account en se basant sur des ID externes,
- le troisième sous-job insère deux enregistrements Account avec leurs ID externes,
- le quatrième insère trois enregistrements Contact en se basant sur la relation de mapping avec les ID externes de l’objet Account et rassemble les données insérées et les données en erreur,
- le cinquième met à jour deux enregistrements Contact insérés, en se basant sur les ID externes de l’objet Account,
- le sixième sous-job récupère les enregistrements Account insérés et
- le septième sous-job récupère les enregistrements Contact sur lesquels un upsert a été effectué.
Créer un Job pour effectuer un upsert sur des données Salesforce en se basant sur des ID externes

Avant de commencer

- Un champ d'ID externe personnalisé Account_External_ID a été ajouté pour l'objet Account dans votre Salesforce, qui sera utilisé pour mapper la relation avec l'objet Contact.
Un champ d’ID externe personnalisé Contact_Talend_ID a été ajouté pour l’objet Contact dans Salesforce, qui sera utilisé en tant que colonne clé pour l’upsert sur les données de l’objet Contact.

**Procédure**

1. Créez un nouveau Job et ajoutez un composant `tSalesforceConnection`, trois `tSalesforceInput`, quatre `tSalesforceOutput`, trois `tFixedFlowInput` et quatre `tLogRow`.
2. Reliez le premier composant `tSalesforceInput` au premier `tSalesforceOutput` à l’aide d’un lien `Row > Main`.
3. Répétez l’opération pour relier le premier `tFixedFlowInput` au deuxième `tSalesforceOutput`, le deuxième `tFixedFlowInput` au troisième `tSalesforceOutput`, le troisième `tFixedFlowInput` au quatrième `tSalesforceOutput`, le deuxième `tSalesforceInput` au troisième `tLogRow` et le troisième `tSalesforceInput` au quatrième `tLogRow`.
4. Reliez le troisième `tSalesforceOutput` au premier `tLogRow` à l’aide d’un lien `Row > Main` et au deuxième `tLogRow` à l’aide d’un lien `Row > Reject`.
5. Reliez le `tSalesforceConnection` au premier `tSalesforceInput`, à l’aide d’un lien `Trigger > On Subjob Ok`.
6. Répétez l’opération pour relier le premier `tSalesforceInput` au premier `tFixedFlowInput`, le premier `tFixedFlowInput` au deuxième `tFixedFlowInput`, le deuxième `tFixedFlowInput` au troisième `tFixedFlowInput`, le troisième `tFixedFlowInput` au deuxième `tSalesforceInput` et le deuxième `tSalesforceInput` au troisième `tSalesforceInput`.

**Ouvrir une connexion à Salesforce**

**Procédure**

1. Double-cliquez sur le `tSalesforceConnection` pour ouvrir sa vue `Basic settings` dans l’onglet Component.
2. Sélectionnez Basic dans la liste déroulante Connection type, afin de vous connecter à Salesforce, à l’aide de votre identifiant de compte Salesforce, son mot de passe et son jeton de sécurité.


3. Dans le champ User Id, saisissez votre identifiant Salesforce.
4. Cliquez sur le bouton [...] à côté du champ Password et, dans la boîte de dialogue qui s’ouvre, saisissez votre mot de passe entre guillemets doubles.
5. Cliquez sur le bouton [...] à côté du champ Security Token et, dans la boîte de dialogue qui s’ouvre, saisissez votre jeton de sécurité Salesforce entre guillemets doubles.
Vous pouvez utiliser l'assistant de métadonnées Salesforce afin de créer une connexion à votre système Salesforce, sauvegarder la connexion dans le référentiel et réutiliser sa configuration définie dans l'assistant de métadonnées, dans les composants Salesforce. Pour plus d’informations, consultez la description concernant la centralisation des métadonnées Salesforce.

**Supprimer des enregistrements Account Salesforce en se basant sur des ID externes**

**Procédure**

1. Double-cliquez sur le premier composant **tSalesforceInput** pour ouvrir sa vue **Basic settings** dans l’onglet **Component**.
2. Sélectionnez le composant créant la connexion à Salesforce, dans la liste déroulante **Connection Component**, le **tSalesforceConnection_1** dans cet exemple.
3. Cliquez sur le bouton « [...] » à côté du champ **Module Name** et, dans la boîte de dialogue, sélectionnez l’objet dont les données seront supprimées, **Account** dans cet exemple. Le schéma de l’objet Account sera automatiquement rempli.
4. Cliquez sur **OK** pour sauvegarder les modifications et, dans la boîte de dialogue qui s’ouvre, cliquez sur **Yes** afin de propager le schéma au composant **tSalesforceOutput**.
5. Sélectionnez **Query** dans la liste déroulante **Query Mode** et spécifiez la condition utilisée pour filtrer les données à supprimer. Dans cet exemple, spécifiez `talendlena__Account_External_ID__c like '%talend%'`.

   *Tous les enregistrements Account Salesforce dont le champ `talendlena__Account_External_ID__c` contient `talend` seront supprimés.*

6. Double-cliquez sur le premier **tSalesforceOutput** pour ouvrir sa vue **Basic settings**, dans l’onglet **Component**.
7. Sélectionnez le composant créant la connexion à Salesforce dans la liste déroulante **Connection Component**, **tSalesforceConnection_1** dans cet exemple.
8. Cliquez sur le bouton « [...] » à côté du champ **Module Name** et, dans la boîte de dialogue qui s’ouvre, sélectionnez l’objet dont les données seront supprimées, **Account** dans cet exemple.
9. Sélectionnez **DELETE** dans la liste déroulante **Output Action** pour effectuer l’opération de suppression sur les objets Account filtrés par le premier **tSalesforceInput**.

**Insérer des enregistrements Account Salesforce avec des ID externes**

**Procédure**

1. Double-cliquez sur le deuxième **tSalesforceOutput** pour ouvrir sa vue **Basic settings**, dans l’onglet **Component**.
2. Sélectionnez le composant créant la connexion à Salesforce dans la liste **Connection Component**, **tSalesforceConnection_1** dans cet exemple.
3. Cliquez sur le bouton « [...] » à côté du champ **Module Name** et, dans la boîte de dialogue qui s’ouvre, sélectionnez l’objet dans lequel les données seront insérées, **Account** dans cet exemple. Le schéma de l’objet Account sera automatiquement rempli.
4. Cliquez sur le bouton « [...] » à côté du champ **Edit schema** et, dans la boîte de dialogue qui s’ouvre, supprimez toutes les colonnes de schéma, à l’exception des deux colonnes **Name** et
Cliquez ensuite sur [copier] pour copier ces deux colonnes dans le premier tFixedFlowInput. Cela fait, cliquez sur OK pour fermer la boîte de dialogue.

5. Sélectionnez INSERT dans la liste déroulante Output Action.


7. Dans la zone Mode, sélectionnez Use Inline Table et, dans la table qui s'affiche, saisissez les données à insérer dans l'objet Account Salesforce. Dans cet exemple, les deux enregistrements suivants seront insérés.

```
account_talend;account_talend_exid
account_talend_doc;account_talend_doc_exid
```

Vous pouvez également sélectionner Use Inline Content, puis copier-coller les données d'entrée dans le champ Content qui s'affiche.

**Insérer des enregistrements Contact Salesforce grâce aux ID externes**

**Procédure**

1. Double-cliquez sur le troisième tSalesforceOutput pour ouvrir sa vue Basic settings, dans l'onglet Component.

2. Sélectionnez le composant créant la connexion à Salesforce dans la liste déroulante Connection Component, tSalesforceConnection_1 dans cet exemple.

3. Cliquez sur le bouton [...] à côté du champ Module Name et, dans la boîte de dialogue qui s'ouvre, sélectionnez l'objet dans lequel les données seront insérées, Contact dans cet exemple. Le schéma de l'objet Contact sera automatiquement renseigné. Cela fait, cliquez sur OK afin de sauvegarder les modifications et, dans la boîte de dialogue qui s'ouvre, cliquez sur Yes pour propager le schéma aux deux composants tLogRow suivants.

4. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la boîte de dialogue qui s'ouvre, supprimez toutes les colonnes du schéma, à l'exception des trois colonnes LastName, FirstName et talendlena__Contact_Talend_ID__c. Ajoutez ensuite une autre colonne Account_External_ID de type String et copiez ces quatre colonnes dans le deuxième composant tFixedFlowInput. Cela
fait, cliquez sur **OK** pour fermer la boîte de dialogue. Dans la boîte de dialogue qui s’ouvre, cliquez sur **Yes** afin de propager le schéma aux deux composants `tLogRow` suivants.

5. Sélectionnez `UPPER` dans la liste déroulante `Output Action` et sélectionnez `talendlena__Contact_Talend_ID__c` dans la liste déroulante `Upsert Key Column` qui s’affiche.

6. Allez dans la vue `Advanced settings` et, dans la table `Relationship mapping for upsert`, spécifiez le mapping de relations pour la colonne `Account_External_ID`, en ajoutant une ligne et configurant la valeur pour chaque colonne de la table. Dans cet exemple, la colonne `Account_External_ID` est mappée au champ d'ID externe `talendlena__Account_External_ID__c` de l'objet `Account`. La valeur de `Column name of Talend Schema` est `Account_External_ID`, la valeur de `Lookup relationship field name` et de `Module name` est `Account` et la valeur de `External id name` est `talendlena__Account_External_ID__c`.

7. Décrochez les cases `Extend Insert` et `Die on Error`, afin de rassembler les données en erreur à l'aide d'un lien `Row > Reject`.


9. Dans la zone `Mode`, sélectionnez `Use Inline Table` et, dans la table qui s’affiche, saisissez les données à insérer dans l’objet `Contact Salesforce`. Dans cet exemple, les données d’entrée sont les suivantes:

```
Beckham;David;2018010001;account_talend_exid
```

9. Dans la zone `Mode`, sélectionnez `Use Inline Table` et, dans la table qui s’affiche, saisissez les données à insérer dans l’objet `Contact Salesforce`. Dans cet exemple, les données d’entrée sont les suivantes:
Vous pouvez également sélectionner **Use Inline Content** et copier-coller les données d’entrée dans le champ **Content** qui s’affiche.

10. Double-cliquez sur le premier **tLogRow** et, dans sa vue **Basic settings**, sélectionnez **Table**, dans la zone **Mode**, pour afficher les résultats sous forme de tableau.

11. Répétez l’opération pour configurer le deuxième **tLogRow**.

**Mettre à jour des enregistrements Contact Salesforce grâce à des ID externes**

**Procédure**

1. Double-cliquez sur le quatrième **tSalesforceOutput** pour ouvrir sa vue **Basic settings** dans l’onglet **Component**.

2. Sélectionnez le composant créant la connexion à Salesforce, dans la liste déroulante **Connection Component**, **tSalesforceConnection_1** dans cet exemple.

3. Cliquez sur le bouton [...] à côté du champ **Module Name** et, dans la boîte de dialogue qui s’ouvre, sélectionnez l’objet dans lequel les données seront insérées, **Contact** dans cet exemple. Le schéma de l’objet Contact sera automatiquement renseigné.

4. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans la boîte de dialogue, supprimez toutes les colonnes du schéma, à l’exception des trois colonnes **Phone**, **Email** et **talendlena__Contact_Talend_ID__c**. Ajoutez une autre colonne **Account_External_ID** de type String et copiez ces quatre colonnes dans le deuxième composant **tFixedFlowInput**. Cela fait, cliquez sur **OK** pour fermer la boîte de dialogue.

5. Sélectionnez **UPsert** dans la liste déroulante **Output Action** et **talendlena__Contact_Talend_ID__c** dans la liste déroulante **Upsert Key Column**.

6. Allez dans la vue **Advanced settings** et, dans la table **Relationship mapping for upsert**, spécifiez le mapping de relation pour la colonne **Account_External_ID**, en ajoutant une ligne et en configurant la valeur pour chaque colonne de la table. Dans cet exemple, la colonne **Account_External_ID** est mappée avec le champ d’ID externe **talendlena__Account_External_ID__c** de l’objet Account. La valeur de **Column name of Talend Schema** est **Account_External_ID**, la valeur de **Lookup relationship field name** et de **Module name** est **Account** et la valeur de **External id name** est **talendlena__Account_External_ID__c**.

7. Double-cliquez sur le troisième composant **tFixedFlowInput** pour ouvrir sa vue **Basic settings** dans l’onglet **Component**.
Dans la zone Mode, sélectionnez Use Inline Table et, dans la table qui s’affiche, saisissez les données utilisées pour mettre à jour les deux enregistrements insérés dans l’objet Contact Salesforce. Dans cet exemple, les données mises à jour se présentent comme suit. Les valeurs des numéros de téléphone et des adresses e-mail seront ajoutées pour les deux enregistrements. La valeur d’ID externe Account pour le second enregistrement est mises à jour en `account_talend_doc_exid`.

```
+86 13666666666;beckham.david@talend.com;2018010001;account_talend_exid
+86 13888888888;taylor.swift@talend.com;2018010002;account_talend_doc_exid
```

Vous pouvez également sélectionner Use Inline Content et copier-coller les données d’entrée dans le champ Content qui s’affiche.

**Récupérer les enregistrements Account Salesforce insérés**

**Procédure**

1. Double-cliquez sur le deuxième tSalesforceInput pour ouvrir sa vue Basic settings dans l’onglet Component.
2. Sélectionnez le composant créant la connexion à Salesforce dans la liste Connection Component, tSalesforceConnection_1 dans cet exemple.
3. Cliquez sur le bouton [...] à côté du champ Module Name et, dans la boîte de dialogue qui s’ouvre, sélectionnez l’objet duquel récupérer les données, Account dans cet exemple. Le schéma de l’objet Account est automatiquement renseigné.
4. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la boîte de dialogue qui s’ouvre, supprimez toutes les colonnes du schéma, à l’exception des trois colonnes `Id`, `Name` et `talendlena__Account_External_ID__c`. Cela fait, cliquez sur OK pour sauvegarder les modifications et, dans la boîte de dialogue qui s’ouvre, cliquez sur Yes pour propager le schéma au composant tLogRow suivant.
5. Sélectionnez Query dans la liste déroulante Query Mode et spécifiez la condition utilisée pour filtrer les données à récupérer. Dans cet exemple, spécifiez Name like '%talend%'.

Tous les enregistrements Account Salesforce dont le champ Name contient talend seront récupérés.
6. Double-cliquez sur le troisième composant tLogRow et, dans sa vue Basic settings, sélectionnez Table, dans la zone Mode, pour afficher les résultats sous forme de tableau.
Récupérer des enregistrements Contact Salesforce mis à jour à l’aide d’une requête SOQL

Procédure

1. Double-cliquez sur le troisième tSalesforceInput pour ouvrir sa vue Basic settings dans l’onglet Component.
2. Sélectionnez le composant créant la connexion à Salesforce, dans la liste Connection Component, tSalesforceConnection_1 dans cet exemple.
3. Cliquez sur le bouton [...] à côté du champ Module Name et, dans la boîte de dialogue qui s’ouvre, sélectionnez l’objet duquel récupérer les données, Contact dans cet exemple. Le schéma de l’objet Contact est automatiquement renseigné.
4. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la boîte de dialogue qui s’ouvre, supprimez toutes les colonnes du schéma, à l’exception des quatre colonnes suivantes: Name, Phone, Email et talendlena__Contact_Talend_ID__c.
5. Ajoutez deux autres colonnes pour récupérer le nom et l’ID externe de l’objet Account lié, Contact_Account_Name et Contact_Account_talendlena__Account_External_ID__c dans cet exemple.

Afin de récupérer les données d’une colonne d’objet liée, il est nécessaire de définir le nom de la colonne en respectant une norme, dans l’éditeur du schéma. La syntaxe correcte est Nomde1ObjetCourant_Nomde1ObjetLié_Nomde1Colonne. Si cette syntaxe n’est pas respectée, les données de l’objet lié ne seront pas retournées. Pour plus d’informations concernant la définition d’un schéma pour une requête de relation, consultez Configurer le schéma pour la fonctionnalité de récupération de requêtes du tSalesforceInput à la page 3613.
6. Cliquez sur OK pour sauvegarder les modifications. Dans la boîte de dialogue qui s’ouvre, cliquez sur Yes afin de propager le schéma au composant tLogRow suivant.
7. Sélectionnez Query dans la liste déroulante Query Mode.
8. Cochez la case Manual Query et cliquez sur le bouton Guess query afin de générer la requête SOQL, en vous basant sur le nom et les colonnes du schéma du module défini. La requête SOQL générée se présente comme suit.

```
SELECT Name, Phone, Email, talendlena__Contact_Talend_ID__c,
```
9. Dans la chaîne de caractères SOQL générée, remplacez chaque tiret bas après le nom de l'objet dans le nom de la colonne `Contact.Account.talendlena__Account_External_ID__c` par un point. Ajoutez la clause de condition WHERE utilisée pour filtrer les données à récupérer à la fin de la requête SOQL générée. Dans cet exemple, ajoutez `talendlena__Contact_Talend_ID__c` like '201801%' pour récupérer tous les enregistrements Contact Salesforce dont le champ `talendlena__Contact_Talend_ID__c` commence par 201801. La requête SOQL mise à jour se présente comme suit.

```sql
SELECT Name, Phone, Email, talendlena__Contact_Talend_ID__c, Contact.Account.Name, Contact.Account.talendlena__Account_External_ID__c FROM Contact
WHERE talendlena__Contact_Talend_ID__c like '201801%'
```

10. Double-cliquez sur le quatrième composant `tLogRow` et, dans sa vue `Basic settings`, sélectionnez `Table` dans la zone `Mode`, pour afficher les résultats sous forme de tableau.

**Exécuter le Job pour effectuer un upsert sur les données de Salesforce en se basant sur un ID externe**

**Procédure**

1. Appuyez sur les touches `Ctrl + S` pour sauvegarder votre Job.
2. Appuyez sur `F6` pour l'exécuter.

Comme affiché ci-dessus, les deux enregistrements `Account account_talend` et `account_talend_doc` ont bien été insérés. Deux enregistrements `Contact Beckham David` et `Taylor Swift` ont bien été insérés et mis à jour à partir de la relation de mapping avec les ID externes dans l’objet `Account` et un autre enregistrement `Contact` dans lequel manque la valeur du champ `LastName` requis n’a pas été inséré.

Vous pouvez également vérifier les résultats d’exécution du Job sur le site Web de Salesforce.
tSalesforceOutputBulk

Ce composant génère le fichier à traiter par le composant tSalesForceBulkExec pour traitement de masse.

Les composants tSalesforceOutputBulk et tSalesforceBulkExec sont utilisés ensemble dans un processus de deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la seconde, ce fichier est utilisé pour alimenter la base de données Salesforce. Ces deux étapes sont fusionnées dans le composant tSalesforceOutputBulkExec. L’avantage d’utiliser deux étapes séparées réside dans le fait que les données peuvent être transformées avant chargement dans la base de données.

Propriétés du tSalesforceOutputBulk Standard

Ces propriétés sont utilisées pour configurer le tSalesforceOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tSalesforceOutputBulk Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
| | • View schema : sélectionnez cette option afin de voir le schéma.
| | • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
| | • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
| | Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent.
| Bulk File Path | Spécifiez l’emplacement du fichier à générer.
| Append | Cochez cette case pour écrire à la suite les nouvelles données à la fin du fichier s’il existe déjà, plutôt que d’écaser les données existantes. |
**Ignore Null**

Cochez cette case pour ignorer les valeurs NULL.

---

**Advanced settings**

**Relationship mapping for upsert**

Cliquez sur le bouton [+] pour ajouter des lignes et spécifiez les champs d'ID externes du flux d'entrée, les champs de relations lookup dans le module d'upsert, le module de lookup ainsi que les champs d'ID externes dans le module de lookup.

- **Column name of Talend Schema**: nom du champ d'ID externe dans le flux d'entrée. Cela fait référence aux champs du schéma du composant précédent. Ces colonnes sont mises en correspondance par rapport aux champs d'ID externes spécifiés dans la colonne **External id name**. Ces champs sont ceux spécifiés dans la colonne **Module name** du module de lookup.

- **Lookup relationship field name**: nom du champ de relations lookup. Cela fait référence aux champs de relations lookup spécifiés dans le champ **Module Name** de la vue **Basic settings** du composant **tSalesforceBulkExec**. Ils servent à établir les relations avec le module de lookup spécifié dans la colonne **Module name** de cette table. Pour plus d'informations concernant la définition des champs de relations lookup et comment saisir le bon nom dans le champ **Lookup field name**, rendez-vous sur le site Web de Salesforce et lancez l’application Salesforce Data Loader.

- **Module name**: nom du module de lookup.


- **External id name**: nom du champ d’ID externe dans le module de lookup spécifié dans la colonne **Module name**.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

---

**Global Variables**

| **NB_LINE** | Nombre de lignes traitées. Cette variable est une variable After et retourne un entier. |
### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec le `tSalesforceBulkExec`. Utilisés ensemble, ils permettent un gain de performance lors de l’alimentation ou de la modification d’informations sur Salesforce.com. |

### Scénario associé

Aucun scénario n’est disponible pour ce composant.
tSalesforceOutputBulkExec

Ce composant charge en masse des données dans un fichier donné, dans un objet Salesforce.

Les composants tSalesforceOutputBulk et tSalesforceBulkExec sont utilisés ensemble dans un processus de deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la seconde, ce fichier est utilisé pour alimenter la base de données Salesforce. Ces deux étapes sont fusionnées dans le composant tSalesforceOutputBulkExec. L'avantage d'utiliser deux étapes séparées réside dans le fait que les données peuvent être transformées avant chargement dans la base de données.

Propriétés du tSalesforceOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tSalesforceOutputBulkExec s'exécutant dans le framework de Jobs Standard.

Le composant tSalesforceOutputBulkExec Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.

| Connection Component | Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant. |

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Sélectionnez dans la liste déroulante le type de connexion. Les propriétés de la connexion varient selon le type de connexion sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Basic</strong> : sélectionnez cette option pour accéder à Salesforce en saisissant votre identifiant et votre mot de passe.</td>
</tr>
<tr>
<td></td>
<td>• <strong>OAuth</strong> : sélectionnez cette option pour accéder à Salesforce via le protocole OAuth (Open Authorization) 2.0 pour authentifier l’utilisateur. Dans la liste déroulante <strong>OAuth2 flow type</strong> qui s’affiche, sélectionnez un type de flux d’authentification OAuth 2.0, <strong>Json Web Token Flow</strong> ou <strong>Implicit</strong></td>
</tr>
</tbody>
</table>
**Flow (Deprecated)** (User Agent Flow). Ainsi, vous devez créer une application connectée dans Salesforce, afin de configurer un fournisseur d’authentification Salesforce.

Pour plus d’informations concernant le flux d’authentification OAuth, consultez [Authenticate Apps with OAuth](en anglais).

Pour plus d’informations concernant la création d’une application connectée dans Salesforce, consultez [Create a Connected App](en anglais).

<table>
<thead>
<tr>
<th>User Id</th>
<th>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Mot de passe Salesforce associé à l’identifiant. Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</td>
</tr>
<tr>
<td>Security Token</td>
<td>Jeton de sécurité Salesforce. Pour plus d’informations, consultez [Reset Your Security Token](en anglais). Cette propriété est disponible uniquement lorsque le type de connexion Basic est sélectionné.</td>
</tr>
<tr>
<td>Issuer</td>
<td>Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez [Create a Connected App](en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>Subject</td>
<td>Identifiant Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>Expiration time (in seconds)</td>
<td>Délai d’expiration de l’assertion (en secondes) durant les cinq prochains minutes. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>Key store password</td>
<td>Mot de passe du Keystore. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Certificate alias</strong></td>
<td>Nom unique du certificat signé par Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Json Web Token Flow est sélectionné.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Client Id</strong></td>
<td>Consumer Key d’OAuth, générée lorsque votre application connectée est créée et affichée dans la page d’informations de l’application connectée dans Salesforce. Pour plus d’informations, consultez <a href="https://%3Clogin%3E.salesforce.com/connectedapp/new">Create a Connected App</a> (en anglais). Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Callback Host</strong></td>
<td>Valeur de l’hôte dans l’URL de la fonction de rappel pour l’authentification OAuth définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Callback Port</strong></td>
<td>Valeur du port dans l’URL de la fonction de rappel pour l’authentification OAuth, définie lors de la création d’une application connectée et qui sera affichée dans la zone API (Enable OAuth Settings) de la page d’informations de l’application connectée dans Salesforce. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Token File</strong></td>
<td>Chemin d’accès au fichier de jeton stockant le jeton de rafraîchissement utilisé pour obtenir le jeton d’accès sans autorisation. Cette propriété est disponible uniquement lorsque le type OAuth Implicit Flow est sélectionné.</td>
</tr>
<tr>
<td><strong>Module Name</strong></td>
<td>Cliquez sur le bouton [...] à côté du champ et, dans la boîte de dialogue qui s’ouvre, sélectionnez le module à utiliser, ou cochez la case Use custom object et spécifiez le nom du module dans le champ Object Name.</td>
</tr>
<tr>
<td><strong>Schema et Edit schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <code>line</code> lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
</tbody>
</table>
• **View schema** : sélectionnez cette option afin de voir le schéma.

• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

### Output Action

<table>
<thead>
<tr>
<th>Sélectionnez dans la liste l’une des opérations suivantes à effectuer.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>INSERT</strong> : insérer un ou plusieurs enregistrements dans Salesforce.</td>
</tr>
<tr>
<td>• <strong>UPDATE</strong> : mettre à jour un ou plusieurs enregistrements existants dans Salesforce.</td>
</tr>
<tr>
<td>• <strong>UPSERT</strong> : créer de nouveaux enregistrements et mettre à jour des enregistrements existants. Dans le champ <strong>Upsert Key Column</strong> affiché, spécifiez la colonne clé pour l’opération d’upsert.</td>
</tr>
<tr>
<td>• <strong>DELETE</strong> : supprimer ou un plusiers enregistrements dans Salesforce.</td>
</tr>
</tbody>
</table>

### Bulk File Path

| Spécifiez le chemin d’accès au fichier stockant les données à traiter. |

### Append

| Cochez cette case pour écrire à la suite les nouvelles données à la fin du fichier s’il existe déjà, plutôt que d’écaser les données existantes. |

### Ignore Null

| Cochez cette case pour ignorer les valeurs NULL. |

### Advanced settings

| **Salesforce URL** | URL de Service Web, requise pour vous connecter à Salesforce. |
| **API version** | Version de l’API Salesforce.  
Cette propriété est disponible uniquement lorsque le type de connexion **OAuth** est sélectionné. |
<p>| <strong>Need compression</strong> | Cochez cette case pour activer la compression des messages SOAP, ce qui peut conduire à améliorer les performances. |
| <strong>Trace HTTP message</strong> | Cochez cette case pour écrire en sortie les interactions HTTP dans la console. |</p>
<table>
<thead>
<tr>
<th>Client Id</th>
<th>Saisissez l’ID d’un utilisateur réel afin de différencier les utilisateurs d’un même compte et mot de passe pour accéder au site Web de Salesforce.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeout</td>
<td>Saisissez, en millisecondes, le délai avant suspension des requêtes dans Salesforce.</td>
</tr>
<tr>
<td>Use Proxy</td>
<td>Cochez cette case pour utiliser un serveur proxy et, dans les champs Host, Port, User Id et Password qui s’affiche, spécifiez les paramètres de connexion du serveur proxy.</td>
</tr>
</tbody>
</table>
| Relationship mapping for upsert | Cliquez sur le bouton [+ ] pour ajouter des lignes et spécifiez les champs d’ID externes du flux d’entrée, les champs de relations lookup dans le module d’upsert, le module de lookup ainsi que les champs d’ID externes dans le module de lookup.  
  - **Column name of Talend Schema** : nom du champ d’ID externe dans le flux d’entrée. Cela fait référence aux champs du schéma du composant précédent. Ces colonnes sont mises en correspondance par rapport aux champs d’ID externes spécifiés dans la colonne **External id name**. Ces champs sont ceux spécifiés dans la colonne **Module name** du module de lookup.  
  - **Lookup relationship field name** : nom du champ de relations lookup. Cela fait référence aux champs de relations lookup spécifiés dans le champ **Module Name** de la vue **Basic settings**. Ils servent à établir les relations avec le module de lookup spécifié dans la colonne **Module name** de cette table. Pour plus d’informations concernant la définition des champs de relations lookup et comment saisir le bon nom dans le champ **Lookup field name**, rendez-vous sur le site Web de Salesforce et lancez l’application Salesforce Data Loader.  
  - **Module name** : nom du module de lookup.  
  - **Polymorphic** : cochez cette case uniquement lorsque les champs polymorphiques sont utilisés pour le mapping de relations. Vous pouvez obtenir une erreur si vous ne cochez pas cette case pour un champ polymorphe. Vous pouvez également obtenir une erreur si vous la sélectionnez pour un champ qui n’est pas polymorphe. Pour plus d’informations concernant les champs polymorphiques, recherchez polymorphic à l’adresse http://www.salesforce.com/us/developer/docs/api_asynch/.  
  - **External id name** : nom du champ d’ID externe dans le module de lookup spécifié dans la colonne **Module name**.  
  Cette propriété est disponible uniquement lorsque l’option **UPsert** est sélectionnée dans la liste déroulante **Output Action**. |
| Concurrency Mode | Sélectionnez le mode de simultanéité du Job.  
  - **Parallel** : traite les lots en mode parallèle. |
### tSalesforceOutputBulkExec

- **Serial** : traite les lots en mode séquentiel.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows to Commit</td>
<td>Spécifiez le nombre de lignes par lot de données à traiter.</td>
</tr>
<tr>
<td>Bytes to Commit</td>
<td>Spécifiez le nombre d’octets par lot de données à traiter.</td>
</tr>
<tr>
<td>Wait Time Check Batch State</td>
<td>Spécifiez le temps d’attente (en millisecondes) avant de vérifier que les lots dans un Job ont été traités, jusqu’à ce que tous les lots soient traités.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Règle d’utilisation</td>
<td>Ce composant est généralement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans Salesforce.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Les données de masse à traiter dans Salesforce.com doivent être au format .csv.</td>
</tr>
</tbody>
</table>

### Insérer des données de masse dans votre Salesforce.com

Ce scénario décrit un Job à quatre composants, chargeant en masse des données dans un fichier, dans Salesforce, effectuant une action sur les données et affichant les résultats d’exécution du Job dans la console.

Le contenu du fichier d’entrée SalesforceAccount.txt utilisé dans cet exemple se présente comme suit :

```
Name;ParentId;Phone;Fax
Burlington Textiles Corp of America;;(336) 222-7000;(336) 222-8000
```
Configurer le Job pour insérer des données en masse dans Salesforce

Procédure

1. Créez un nouveau Job et ajoutez un tFileInputDelimited, un tSalesforceOutputBulkExec et deux tLogRow en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.

2. Reliez le tFileInputDelimited au tSalesforceOutputBulkExec à l’aide d’un lien Row > Main.

3. Reliez le tSalesforceOutputBulkExec au premier tLogRow à l’aide d’un lien Row > Main.

4. Reliez le tSalesforceOutputBulkExec au second tLogRow à l’aide d’un lien Row > Reject.

Configurer le Job pour insérer des données de masse dans Salesforce

Procédure

1. Double-cliquez sur le tFileInputDelimited pour ouvrir sa vue Basic settings.


3. Cliquez sur le bouton [...] à côté du champ Edit schema et dans l’éditeur, définissez le schéma en ajoutant quatre colonnes Name, ParentId, Phone et Fax de type String. Cela fait, cliquez sur OK pour sauvegarder les modifications et fermer la boîte de dialogue.
4. Double-cliquez sur le composant **tSalesforceOutputBulkExec** pour ouvrir sa vue **Basic settings**.

5. Dans les champs **User Id**, **Password** et **Security Key**, saisissez les informations d’authentification de l’utilisateur requises pour accéder à Salesforce.

6. Cliquez sur le bouton [...] à côté du champ **Module Name** et, dans la boîte de dialogue qui s’ouvre, sélectionnez l’objet auquel accéder. Dans cet exemple, sélectionnez **Account**.

7. Dans le champ **Bulk File Path**, parcourrez votre système ou saisissez le chemin d’accès au fichier CSV stockant les données pour traitement de masse. Le fichier de masse à traiter doit être au format CSV.

8. Double-cliquez sur le premier **tLogRow** pour ouvrir sa vue **Basic settings**.
Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour une lisibilité optimale des résultats.

**10. Répétez l’opération pour configurer le second tLogRow.**

### Exécuter le Job pour insérer des données en masse dans Salesforce

**Procédure**

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.

Dans la console de la vue **Run**, vous pouvez vérifier les résultats d’exécution.

Dans le tableau **tLogRow_1**, vous pouvez lire les données insérées dans Salesforce.

Dans le tableau **tLogRow_2**, vous pouvez lire les données rejetées à cause d’une incompatibilité avec les objets Account auxquels vous avez accédé.

Si vous souhaitez transformer les données d’entrée avant de les charger dans Salesforce, vous devez utiliser les composants **tSalesforceOutputBulk** et **tSalesforceBulkExec** en coopération.
tSalesforceWaveBulkExec

Ce composant charge des données dans Salesforce Analytics Cloud à partir d'un fichier local.

Propriétés du tSalesforceWaveBulkExec Standard

Ces propriétés sont utilisées pour configurer le tSalesforceWaveBulkExec s'exécutant dans le framework de Jobs Standard.

Le composant tSalesforceWaveBulkExec Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-In</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste **Component List** présente uniquement les composants de connexion du Job du même niveau.

**User Name** et **Password**

Saisissez les détails d’authentification du service Web. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**End Point**

Saisissez l’URL du WebService nécessaire pour se connecter à Salesforce, https://login.salesforce.com/services/Soap/u/37.0 par exemple. Notez que la version dans l’URL doit être 32.0 ou ultérieure.

**Schema** et **Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

- Pour les champs de type numérique (par exemple **byte**, **short**, **int**, **long**, **float**, **double** et **BigDecimal**), vous devez spécifier leur valeurs de longueur et de précision, dans l’éditeur de schéma. Les valeurs par défaut de longueur et précision sont respectivement
10 et 2, mais vous pouvez spécifier des valeurs personnalisées dans l’éditeur du schéma.

- Pour les champs de type date, vous devez spécifier le format de la date dans l’éditeur de schéma. Pour plus d’informations concernant les formats de date supportés, consultez [Analytics Cloud External Data Format Reference](en anglais).

La liste **Schema** et le bouton **Editor schema** ne sont pas disponibles pas si la case **Custom JSON Metadata** est cochée.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le [Guide utilisateur du Studio Talend](.)

**Repository** : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le [Guide utilisateur du Studio Talend](.).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Operation**

 Sélectionnez l’opération à effectuer sur l’ensemble de données :

- **Append** : Ajoute toutes les données à l’ensemble de données. Crée un ensemble de données s’il n’existe pas.
- **Upsert** : Insère ou met à jour des lignes dans l’ensemble de données. Crée un ensemble de données s’il n’existe pas.
- **Overwrite** : Crée un nouvel ensemble de données à partir des données fournies et remplace l’ensemble de données s’il existe déjà.
- **Delete** : Supprime les lignes de l’ensemble de données.

**Remarque** :

- Un fichier de métadonnées JSON est requis pour les opérations **Append**, **Upsert** et **Delete**.
• Les données et les métadonnées des opérations Append et Upsert doivent correspondre à l’ensemble de données sur lequel l’opération est effectuée. Toutes les colonnes, les dimensions et les mesures doivent correspondre parfaitement.

• L’opération Append n’est pas autorisée si vous définissez n’importe quelle colonne comme étant la clé primaire.

• Vous devez spécifier une (et seulement une) colonne en tant que clé primaire sur laquelle l’opération Upsert ou Delete est basée. Vous pouvez le faire en cliquant sur Edit schema et en cochant la case située à côté de la colonne que vous souhaitez définir en tant que clé primaire.

• Les métadonnées pour l’opération Delete doivent être un sous-ensemble des colonnes de l’ensemble de données.

<table>
<thead>
<tr>
<th>Name</th>
<th>Saisissez le nom de l’ensemble de données dans lequel charger les données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV File</td>
<td>Spécifiez le chemin vers le fichier CSV à charger.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>CSV Encoding</th>
<th>Saisissez le type d’encodage du fichier CSV. Notez que la valeur de ce champ doit être la même que celle définie dans le fichier de métadonnées JSON lorsque la case Custom JSON Metadata est cochée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields Delimiter</td>
<td>Saisissez le caractère séparant les valeurs des champs dans le fichier CSV. Notez que la valeur de ce champ doit être la même que celle définie dans le fichier de métadonnées JSON lorsque la case Custom JSON Metadata est cochée.</td>
</tr>
<tr>
<td>Fields Enclosed By</td>
<td>Saisissez le caractère utilisé pour entourer les valeurs des champs dans le fichier CSV. Notez que la valeur de ce champ doit être la même que celle définie dans le fichier de métadonnées JSON lorsque la case Custom JSON Metadata est cochée.</td>
</tr>
<tr>
<td>Line Terminated By</td>
<td>Saisissez le caractère indiquant la fin d’une ligne. Notez que la valeur de ce champ doit être la même que celle définie dans le fichier de métadonnées JSON lorsque la case Custom JSON Metadata est cochée.</td>
</tr>
<tr>
<td>Auto Generate JSON Metadata Description</td>
<td>Cochez cette case pour générer automatiquement la description des métadonnées JSON.</td>
</tr>
<tr>
<td>Header</td>
<td>Spécifiez le nombre de lignes à ignorer dans le fichier CSV.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Unique API Name</td>
<td>Spécifiez le nom unique de l’API pour l’objet dans la description des métadonnées JSON.</td>
</tr>
<tr>
<td>Label</td>
<td>Spécifiez le nom à afficher pour l’objet dans la description des métadonnées JSON.</td>
</tr>
<tr>
<td>Fully Qualified Name</td>
<td>Spécifiez le chemin complet qui identifie de manière unique l’enregistrement dans la description des métadonnées JSON.</td>
</tr>
<tr>
<td>Custom JSON Metadata</td>
<td>Cochez cette case pour utiliser un fichier de métadonnées JSON personnalisé.</td>
</tr>
<tr>
<td>JSON Metadata</td>
<td>Spécifiez le chemin vers le fichier de métadonnées JSON personnalisé.</td>
</tr>
<tr>
<td>Generate JSON in File</td>
<td>Cochez cette case pour écrire la description des métadonnées JSON dans un fichier local.</td>
</tr>
<tr>
<td>Generated JSON Folder</td>
<td>Spécifiez le répertoire dans lequel vous souhaitez stocker le fichier de métadonnées JSON générée.</td>
</tr>
<tr>
<td>Retrieve Upload Status</td>
<td>Cochez cette case pour récupérer le statut du chargement de données.</td>
</tr>
<tr>
<td>Time to wait for server answer (seconds)</td>
<td>Spécifiez, en secondes, le temps d’attente de la réponse du statut de chargement provenant du serveur.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>
Global Variables

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en standalorne. |

Scénario associé

Aucun scénario n’est disponible pour ce composant.
**tSalesforceWaveOutputBulkExec**

Ce composant améliore les performances durant les opérations de données sur Salesforce Analytics Cloud.

Le tSalesforceWaveOutputBulkExec reçoit des données depuis le composant précédent, génère un fichier CSV local, puis charge les données de ce fichier sur Salesforce Analytics Cloud.

**Propriétés du tSalesforceWaveOutputBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tSalesforceWaveOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tSalesforceWaveOutputBulkExec Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

|---------------|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, la liste Component List présente uniquement les composants de connexion du Job du même niveau.

**User Name et Password**

Saisissez les détails d’authentification du service Web. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**End Point**

Saisissez l’URL du WebService nécessaire pour se connecter à Salesforce, https://login.salesforce.com/services/Soap/u/37.0 par exemple. Notez que la version dans l’URL doit être 32.0 ou ultérieure.

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

- Pour les champs de type numérique (par exemple byte, short, int, long, float, double et BigDecimal),
vous devez spécifier leur valeurs de longueur et de précision dans l’éditeur de schéma. Les valeurs par défaut de longueur et précision sont respectivement 10 et 2, mais vous pouvez spécifier des valeurs personnalisées dans l’éditeur du schéma.

- Pour les champs de type date, vous devez spécifier le format de la date dans l’éditeur de schéma. Pour plus d’informations concernant les formats de date supportés, consultez Analytics Cloud External Data Format Reference (en anglais).

La liste Schema et le bouton Edit schema ne sont pas disponibles lorsque la case Custom JSON Metadata est cochée.

| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent.

| Operation |
| Sélectionnez l’opération à effectuer sur l’ensemble de données :
- **Append** : Ajoute toutes les données à l’ensemble de données. Crée un ensemble de données s’il n’existe pas.
- **Upsert** : Insère ou met à jour des lignes dans l’ensemble de données. Crée un ensemble de données s’il n’existe pas.
- **Overwrite** : Crée un nouvel ensemble de données à partir des données fournies et remplace l’ensemble de données s’il existe déjà.
- **Delete** : Supprime les lignes de l’ensemble de données. |
Remarque :

- Un fichier de métadonnées JSON est requis pour les opérations Append, Upsert et Delete.

- Les données et les métadonnées des opérations Append et Upsert doivent correspondre à l’ensemble de données sur lequel l’opération est effectuée. Toutes les colonnes, les dimensions et les mesures doivent correspondre parfaitement.

- L’opération Append n’est pas autorisée si vous définissez n’importe quelle colonne comme étant la clé primaire.

- Vous devez spécifier une (et seulement une) colonne en tant que clé primaire sur laquelle l’opération Upsert ou Delete est basée. Vous pouvez le faire en cliquant sur Edit schema et en cochant la case située à côté de la colonne que vous souhaitez définir en tant que clé primaire.

- Les métadonnées pour l’opération Delete doivent être un sous-ensemble des colonnes de l’ensemble de données.

---

Name

Saisissez le nom de l’ensemble de données dans lequel charger les données.

Generated Temp CSV File

Spécifiez le chemin vers le fichier CSV à générer.

Append

Cochez cette case pour écrire à la suite les données dans le fichier local spécifié s’il existe au lieu de l’écraser.

Include Header

Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.

---

Advanced settings

CSV Encoding

Saisissez le type d’encodage du fichier CSV.
Notez que la valeur de ce champ doit être la même que celle définie dans le fichier de métadonnées JSON lorsque la case Custom JSON Metadata est cochée.

Fields Delimiter

Saisissez le caractère séparant les valeurs des champs dans le fichier CSV.
Notez que la valeur de ce champ doit être la même que celle définie dans le fichier de métadonnées JSON lorsque la case Custom JSON Metadata est cochée.

Fields Enclosed By

Saisissez le caractère utilisé pour entourer les valeurs des champs dans le fichier CSV.
Notez que la valeur de ce champ doit être la même que celle définie dans le fichier de métadonnées JSON lorsque la case Custom JSON Metadata est cochée.
<table>
<thead>
<tr>
<th>Line Terminated By</th>
<th>Saisissez le caractère indiquant la fin d’une ligne. Notez que la valeur de ce champ doit être la même que celle définie dans le fichier de métadonnées JSON lorsque la case Custom JSON Metadata est cochée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create directory if not exists</td>
<td>Cochez cette case pour créer, s’il n’existe pas, le répertoire spécifié dans le champ Generated Temp CSV File.</td>
</tr>
<tr>
<td>Custom the flush buffer size</td>
<td>Cochez cette case pour spécifier le nombre de lignes à écrire avant de vider la mémoire tampon.</td>
</tr>
<tr>
<td>Row number</td>
<td>Spécifiez le nombre de lignes à écrire avant de vider la mémoire tampon. Ce champ est disponible uniquement lorsque la case Custom the flush buffer size est cochée.</td>
</tr>
<tr>
<td>Don’t generate empty file</td>
<td>Cochez cette case si vous ne souhaitez pas générer de fichier vide.</td>
</tr>
<tr>
<td>Auto Generate JSON Metadata Description</td>
<td>Cochez cette case pour générer automatiquement la description des métadonnées JSON.</td>
</tr>
<tr>
<td>Header</td>
<td>Spécifiez le nombre de lignes à ignorer dans le fichier CSV. Ce champ est disponible uniquement lorsque la case Auto Generate JSON Metadata Description n’est pas cochée.</td>
</tr>
<tr>
<td>Unique API Name</td>
<td>Spécifiez le nom unique de l’API pour l’objet dans la description des métadonnées JSON. Ce champ est disponible uniquement lorsque la case Auto Generate JSON Metadata Description n’est pas cochée.</td>
</tr>
<tr>
<td>Label</td>
<td>Spécifiez le nom à afficher pour l’objet dans la description des métadonnées JSON. Ce champ est disponible uniquement lorsque la case Auto Generate JSON Metadata Description n’est pas cochée.</td>
</tr>
<tr>
<td>Fully Qualified Name</td>
<td>Spécifiez le chemin complet qui identifie de manière unique l’enregistrement dans la description des métadonnées JSON. Ce champ est disponible uniquement lorsque la case Auto Generate JSON Metadata Description n’est pas cochée.</td>
</tr>
<tr>
<td>Custom JSON Metadata</td>
<td>Cochez cette case pour utiliser un fichier de métadonnées JSON personnalisé. Cette case est disponible uniquement lorsque la case Auto Generate JSON Metadata Description n’est pas cochée.</td>
</tr>
</tbody>
</table>
### JSON Metadata

Spécifiez le chemin vers le fichier de métadonnées JSON personnalisé.

Ce champ est disponible uniquement lorsque la case Custom JSON Metadata est cochée.

### Generate JSON in File

Cochez cette case pour écrire la description des métadonnées JSON dans un fichier local.

Cette case n’est pas disponible lorsque la case Custom JSON Metadata est cochée.

### Generated JSON Folder

Spécifiez le répertoire dans lequel vous souhaitez stocker le fichier de métadonnées JSON généré.

Ce champ est disponible uniquement lorsque la case Generate JSON in File est cochée.

### Retrieve Upload Status

Cochez cette case pour récupérer le statut du chargement de données.

### Time to wait for server answer (seconds)

Spécifiez, en secondes, le temps d’attente de la réponse du statut de chargement provenant du serveur.

Ce champ est disponible uniquement lorsque la case Retrieve Upload Status.

### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>ERROR_MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger sur Salesforce Analytics Cloud.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé, consultez Insérer des données de masse dans votre Salesforce.com à la page 3644.
**tSampleRow**

Ce composant sélectionne des lignes en fonction d’une liste de lignes simples et/ou de groupes de lignes.
Le tSampleRow filtre les lignes en fonction de leur position.

**Propriétés du tSampleRow Standard**

Ces propriétés sont utilisées pour configurer le tSampleRow s’exécutant dans le framework de Jobs Standard.
Le composant tSampleRow Standard appartient à la famille Processing.
Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
Cliquez sur Sync columns pour récupérer le schéma du composant précédent. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Renseignez la position ou l’intervalle choisi, en utilisant la syntaxe appropriée, pour récupérer une liste de lignes simples et/ou de groupes de lignes.</td>
</tr>
</tbody>
</table>
Global Variables

Global Variables

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant traite des flux de données, il nécessite donc des composants d’entrée et sortie.

Scénario : Filtrer des lignes et des groupes de lignes

Ce scénario Java décrit un Job à trois composants. On utilise un composant tRowGenerator pour créer des entrées au hasard qui sont directement envoyées au tSampleRow, où elles seront filtrées en fonction de leur position. Dans ce scénario, on suppose que le flux d’entrée contient les noms des vendeurs associés au nombre de produits qu’ils ont chacun vendus ainsi que le nombre d’années passées dans l’entreprise. Le résultat de l’opération de filtrage apparaît dans la console Run.

Déposer et relier les composants

Procédure

1. A partir de la Palette, cliquez déposez les composants suivants dans l’éditeur graphique : le tRowGenerator, le tSampleRow et le tLogRow.
2. Connectez ces trois composants à l’aide de liens de type Row > Main.
Résultats

Configurer les composants

Procédure

1. Dans l’éditeur graphique, sélectionnez le tRowGenerator et cliquez sur la vue Component pour définir la configuration de base (Basic settings) du tRowGenerator.
2. Cliquez sur le bouton [...] à côté du champ Edit Schema pour définir les données que vous voulez utiliser en entrée. Dans ce scénario, le schéma est composé de cinq colonnes.
3. Dans l’onglet Basic settings, cliquez sur RowGenerator Editor pour définir les données à générer.
4. Dans l’éditeur RowGenerator Editor, précisez le nombre de lignes à générer dans le champ Number of Rows for RowGenerator puis cliquez sur OK pour fermer l’éditeur.
5. Dans l’éditeur graphique, sélectionnez le tSampleRow et cliquez sur la vue Component pour définir la configuration de base (Basic settings) du tSampleRow.
Dans l’onglet **Basic settings**, définissez l’option **Schema** en mode **Built-In** puis cliquez sur **Sync columns** pour récupérer le schéma à partir du composant **tRowGenerator**.

Dans la boîte de texte **Range**, définissez le filtre de sélection des lignes en utilisant la syntaxe appropriée comme indiqué plus haut. Dans ce scénario, on veut sélectionner la première et la cinquième ligne, ainsi que le groupe de lignes entre 9 et 12.

Dans l’espace graphique, sélectionnez le **tLogRow** puis cliquez sur la vue **Component** pour en définir la configuration de base (**Basic settings**). Pour plus d’informations sur les propriétés du **tLogRow**, consultez tLogRow à la page 2105.

### Sauvegarder et exécuter le Job

**Procédure**

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** ou cliquez sur le bouton **Run** dans la vue **Run** pour exécuter le Job.

**Résultats**

<table>
<thead>
<tr>
<th>ID</th>
<th>Years_in_Empire</th>
<th>First_Name</th>
<th>Last_Name</th>
<th>Select_Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Martin</td>
<td>Taft</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Franklin</td>
<td>Adams</td>
<td>45</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>William</td>
<td>Madison</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>George</td>
<td>Kennedy</td>
<td>47</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>George</td>
<td>Van Buren</td>
<td>16</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>Theodore</td>
<td>Grant</td>
<td>68</td>
</tr>
</tbody>
</table>

*Job sample_Row ended at 11:16 20/08/2008. [exit code=0]*

Les résultats filtrés apparaissent dans la console de log. Ils affichent la première et la cinquième ligne, ainsi que les lignes entre 9 et 12.
tSAPHanaClose

Ce composant ferme une connexion à la base de données SAP Hana.

Propriétés du tSAPHanaClose Standard

Ces propriétés sont utilisées pour configurer le tSAPHanaClose s'exécutant dans le framework de Jobs Standard.

Le composant tSAPHanaClose Standard appartient à la famille Databases.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

Remarque :

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tSAPHanaConnection dans la liste si plus d'une connexion est définie dans le Job.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour connecter les données de log au niveau de chaque composant. |

Utilisation

| Règle d'utilisation | Ce composant est généralement utilisé avec d'autres composants SAP Hana, notamment le tSAPHanaConnection et le tSAPHanaCommit. |

Scénario associé

Aucun scénario n'est disponible pour la version Standard de ce composant.
**tSAPHanaCommit**

Ce composant committe en une seule fois une transaction globale en utilisant une connexion unique, au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

Le tSAPHanaCommit valide les données traitées dans un Job à partir d’une base de données connectée.

**Propriétés du tSAPHanaCommit Standard**

Ces propriétés sont utilisées pour configurer le tSAPHanaCommit s’exécutant dans le framework de Jobs Standard.

Le composant tSAPHanaCommit Standard appartient à la famille Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant <strong>tSAPHanaConnection</strong> dans la liste si plus d’une connexion est définie dans le Job.</td>
</tr>
</tbody>
</table>

**Close Connection**

Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.

⚠️ **Avertissement :**

Si vous utilisez un lien de type **Row > Main** pour relier le **tSAPHanaCommit** à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case **Close connection** car la connexion sera fermée avant la fin du commit de votre première ligne.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour connecter les données de log au niveau de chaque composant. |
**Utilisation**

| Règle d'utilisation | Ce composant est généralement utilisé avec d'autres composants SAP Hana, notamment avec le tSAPHanaConnection et le tSAPHanaRollback. Utilisez ce composant si l'option Auto Commit du tSAPHanaConnection est décochée. |

**Scénario associé**

Pour un scénario associé à ce composant, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
**tSAPHanaConnection**

Ce composant établit une connexion à SAP Hana à réutiliser dans d'autres composants SAP Hana au sein du Job.

Le tSAPHanaConnection ouvre une connexion à la base de données SAP Hana afin d'effectuer une transaction.

**Propriétés du tSAPHanaConnection Standard**

Ces propriétés sont utilisées pour configurer le tSAPHanaConnection s'exécutant dans le framework de Jobs Standard.

Le composant tSAPHanaConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

⚠️ **Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Version</td>
<td>Sélectionnez la version de la base de données SAP Hana Database (HDB) que vous utilisez.</td>
</tr>
<tr>
<td>Host</td>
<td>Saisissez l’adresse IP du serveur de la base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute du serveur de la base de données.</td>
</tr>
<tr>
<td>Table Schema</td>
<td>Saisissez le nom du schéma de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez vos informations d'authentification à la base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Additional JDBC Parameters</td>
<td>Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.</td>
</tr>
</tbody>
</table>
**Advanced settings**

| **Auto Commit** | Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.  
Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.  
Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne committe pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log lors du traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Utilisation**

| **Règle d’utilisation** | Ce composant est généralement utilisé avec d’autres composants SAP Hana, notamment le tSAPHanaCommit et le tSAPHanaRollback, par exemple pour fermer une connexion à la fin d’une transaction. |

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tSAPHanaInput

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma.

Le tSAPHanaInput lit une base de données et extrait des champs selon une requête. Les champs récupérés sont ensuite transmis au composant suivant via une connexion de flux Main > Row.

Propriétés du tSAPHanaInput Standard

Ces propriétés sont utilisées pour configurer le tSAPHanaInput s’exécutant dans le framework de Jobs Standard.

Le composant tSAPHanaInput Standard appartient à la famille Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existante entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.
<table>
<thead>
<tr>
<th><strong>DB Version</strong></th>
<th>Sélectionnez la version de SAP Hana Database (HDB) que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Saisissez l’adresse IP du serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Saisissez le numéro du port d’écoute du serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Nom du schéma de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez vos informations d’authentification à la base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

**Built-in** : Le schéma sera créé et conservé pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

**Repository** : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

<table>
<thead>
<tr>
<th><strong>Table Name</strong></th>
<th>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query Type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
</tbody>
</table>
**Built-in** : Saisissez manuellement l’instruction de requête ou construisez-la graphiquement via le SQL Builder.

**Repository** : Sélectionnez la requête stockée dans le Repository. Le champ Query est automatiquement renseigné.

**Guess Query**
Cliquez sur le bouton **Guess Query** afin de générer la requête correspondant au schéma de votre table dans le champ Query.

**Guess schema**
Cliquez sur le bouton **Guess schema** pour récupérer le schéma de la table.

**Query**
Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

### Advanced settings

**Additional JDBC Parameters**
Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Cette option est indisponible lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**.

**Trim all the String/Char columns**
Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes de type String/Char.

**Trim column**
Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.

### Remarque :
Décrochez la case **Trim all the String/Char columns** pour activer cette option.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau de chaque composant.

### Global Variables

**Global Variables**

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

**QUERY** : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en standalone car il comprend le moteur de SAP Hana. C’est un composant de début pouvant initier un traitement de flux de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tSAPHanaOutput

Ce composant exécute l’action définie sur la table et/ou sur les données contenues dans la table, à partir du flux entrant du composant précédent dans le Job.

Le tSAPHanaOutput écrit, met à jour, modifie ou supprime des entrées dans une base de données SAP Hana.

Propriétés du tSAPHanaOutput Standard

Ces propriétés sont utilisées pour configurer le tSAPHanaOutput s’exécutant dans le framework de Jobs Standard.

Le composant tSAPHanaOutput Standard appartient à la famille Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.
<table>
<thead>
<tr>
<th><strong>DB Version</strong></th>
<th>Sélectionnez la version de SAP Hana Database (HDB) que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Saisissez l’adresse IP du serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Saisissez le numéro du port d’écoute du serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Table Schema</strong></td>
<td>Saisissez le nom du schéma de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez vos informations d’authentification à la base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Saisissez le nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois.</td>
</tr>
<tr>
<td><strong>Action on table</strong></td>
<td>Sur la table définie, vous pouvez effectuer l’une des opérations suivantes :</td>
</tr>
<tr>
<td></td>
<td><strong>None</strong> : Aucune opération n’est effectuée.</td>
</tr>
<tr>
<td></td>
<td><strong>Drop and create table</strong> : La table est supprimée et créée à nouveau.</td>
</tr>
<tr>
<td></td>
<td><strong>Create table</strong> : La table n’existe pas et est créée.</td>
</tr>
<tr>
<td></td>
<td><strong>Create table if not exists</strong> : La table est créée si elle n’existe pas.</td>
</tr>
<tr>
<td></td>
<td><strong>Drop table if exists and create</strong> : La table est supprimée si elle existe déjà puis créée à nouveau.</td>
</tr>
<tr>
<td></td>
<td><strong>Clear table</strong> : Le contenu de la table est supprimé.</td>
</tr>
<tr>
<td><strong>Type of table storage</strong></td>
<td>Cette option n’est disponible que si vous créez (en supprimant ou pas) la table de la base de données. Cette option vous permet de définir la manière dont les données sont stockées dans la table. Les types d’organisation du stockage des données suivants sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>Row</strong> : Les données sont stockées sous forme de lignes. Il est préférable d’utiliser ce type de table si la majorité des accès à la table consiste en la sélection de quelques enregistrements avec tous les attributs sélectionnés.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Column</strong> : Les données sont stockées sous forme de colonnes. Il est préférable d’utiliser ce type de table si la majorité des accès à la table s’effectue un grand nombre de fois avec seulement quelques attributs sélectionnés.</td>
</tr>
<tr>
<td></td>
<td>• <strong>History column</strong> : Une table avec un type de session de transaction particulier, HISTORY est créée. Les tables avec le type de session HISTORY supportent les requêtes de type time travel. Pour plus d’informations concernant HISTORY et les requêtes</td>
</tr>
</tbody>
</table>

- **Global temporary**: La définition de la table et les données peuvent être vues par tous les utilisateurs et toutes les connexions pendant la durée de la session en cours. La table est supprimée à la fin de la session.
- **Local temporary**: La définition de la table et les données ne peuvent être vues que par l’utilisateur qui a créé la table pendant la durée de la session en cours. La table est supprimée à la fin de la session.

### Action on data

Sur les données de la table définie, vous pouvez choisir d'effectuer l'opération suivante :

- **Insert**: Ajouter de nouvelles entrées à la table. Si des doublons sont trouvés, le Job s'arrête.
- **Update**: Apporter des modifications à des entrées existantes.
- **Insert or update**: insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Update or insert**: met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
- **Delete**: Supprimer des entrées correspondant au flux d’entrée.

**Avertissement** :

Il est nécessaire de spécifier au moins une colonne en tant que clé primaire sur laquelle baser les opérations Update et Delete. Vous pouvez faire ceci en cliquant sur *Edit Schema* et en cochant la case correspondant à la colonne (aux colonnes) que vous souhaitez définir comme clé primaire. Pour une utilisation avancée, consultez la description de l’option *Use field options*, dans l’onglet *Advanced settings* de ce composant.

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.

**Built-In**: Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

**Repository**: Le schéma existe déjà et il est stocké dans le *Repository*. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces
 Valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Die on error**

Cette case est décochée par défaut, afin d'ignorer les lignes en erreur et de terminer le traitement avec les lignes sans erreur. Si nécessaire, vous pouvez récupérer les lignes en erreur via un lien **Row > Rejects**.

### Advanced settings

**Additional JDBC Parameters**

Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Cette option est indisponible lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**.

**Commit every**

Saisissez le nombre de lignes à terminer avant de commuter des lots de lignes en même temps dans la base de données. Cette option vous assure une certaine qualité de transaction (mais pas de rollback) et, surtout, une meilleure performance à l'exécution.

**Additional Columns**

Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.

- **Name** : Saisissez le nom de la colonne à modifier ou à insérer.

- **SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.
Position : Sélectionnez Before, Replace ou After, en fonction de l’action à effectuer sur la colonne de référence.

Reference column : Saisissez une colonne de référence que le composant tSAPHanaOutput peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

Use field options

Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données. Dans cette table, vous pouvez :

Key in update : définir la clé primaire à utiliser lors d’une opération de mise à jour.

Key in delete : définir la clé primaire à utiliser lors d’une opération de suppression.

Updatable : définir les colonnes à mettre à jour.

Insertable : définir les colonnes à insérer.

Enable debug mode

Cochez cette case pour afficher chaque étape durant le traitement des entrées dans une base de données.

Support null in “SQL WHERE” statement

Cochez cette case pour prendre en compte les valeurs nulls dans une instruction “SQL WHERE”.

Use Batch Size

Lorsque cette case est cochée, saisissez le nombre de lignes contenues dans chaque lot traité. Cette option est disponible uniquement lorsque la case Use an existing connection n’est pas cochée dans la vue Basic settings.

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

Global Variables

NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.

NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.

NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.

NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.

QUERY : requête traitée. Cette variable est une variable After et retourne une chaîne de caractères.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est
une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

#### Règle d’utilisation

Ce composant doit être relié à un composant d’entrée. Il vous permet d’effectuer des actions sur une table ou sur les données d’une de la base de données SAP Hana. Cela vous permet de créer des flux de rejet à l’aide d’un lien **Row > Reject** afin de filtrer les données en erreur.

#### Dynamic settings

Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641** et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520**. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tSAPHanaRollback

Ce composant évite le commit de transaction involontaire.
Le tSAPHanaRollback annule une transaction dans la base de données connectée.

Propriétés du tSAPHanaRollback Standard

Ces propriétés sont utilisées pour configurer le tSAPHanaRollback s’exécutant dans le framework de Jobs Standard.
Le composant tSAPHanaRollback Standard appartient à la famille Databases.
Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez dans la liste le composant tSAPHanaConnection si plus d’une connexion est définie dans le Job.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion sélectionnée une fois que le composant a terminé ses tâches.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics   | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

| Règle d’utilisation       | Ce composant est généralement utilisé avec d’autres composants SAP Hana, notamment le tSAPHanaConnection et tSAPHanaCommit, afin de mener à bien des transactions. |
Scénario associé

Aucun scénario n'est disponible pour la version Standard de ce composant.
**tSAPHanaRow**

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler).

Le tSAPHanaRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

**Propriétés du tSAPHanaRow Standard**

Ces propriétés sont utilisées pour configurer le tSAPHanaRow s’exécutant dans le framework de Jobs Standard.

Le composant tSAPHanaRow Standard appartient à la famille Databases.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux...
<table>
<thead>
<tr>
<th><strong>DB Version</strong></th>
<th>Sélectionnez la version de SAP Hana Database (HDB) que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
<td></td>
</tr>
<tr>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
<td></td>
</tr>
<tr>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
<td></td>
</tr>
<tr>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois.</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Query Type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.</td>
</tr>
<tr>
<td>Guess Query</td>
<td>Cliquez sur le bouton Guess Query afin de générer la requête correspondant au schéma de votre table dans le champ Query.</td>
</tr>
<tr>
<td>Query</td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décchez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Rejects.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>Additional JDBC Parameters</th>
<th>Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données. Cette option est indisponible si vous avez coché la case Use an existing connection dans la vue Basic settings.</th>
</tr>
</thead>
</table>
| Propagate QUERY’s recordset| Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column.  

1. **Remarque :**
   Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type Object. Ce composant est généralement suivi du tParseRecordSet. |
| Commit every              | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution. |
**Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL.

**Parameter Type** : Saisissez le type du paramètre.

**Parameter Value** : Saisissez la valeur du paramètre.

**Remarque** :
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**QUERY** : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

**Règle d’utilisation**
Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.

**Scénario associé**
Aucun scénario n’est disponible pour la version Standard de ce composant.
tSasInput

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma.

⚠️ Avertissement :

Assurez-vous d’installer au préalable les modules sas.core.jar, sas.intrnet.javatools.jar et sas.svc.connection.jar sous le chemin lib > java dans votre build du Studio Talend pour pouvoir utiliser toutes les fonctionnalités des composants SAS. Vous pouvez vérifier que ces modules ont bien été installés à partir de la vue Modules de votre Studio Talend.

Le tSasInput lit une base de données et en extrait des champs à l’aide de requêtes. Les champs récupérés sont ensuite transmis au composant suivant via une connexion de flux (Main row).

Propriétés du tSasInput Standard

Ces propriétés sont utilisées pour configurer le tSasInput s’exécutant dans le framework de Jobs Standard.

Le composant tSasInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>🔄</td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant. Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Host name</td>
<td>Nom ou adresse IP du serveur SAS.</td>
</tr>
<tr>
<td>Port</td>
<td>Port d’écoute à ouvrir.</td>
</tr>
</tbody>
</table>
| Librefs       | Renseignez le nom du répertoire physique dans lequel est contenue la table à lire, suivi de son chemin d’accès. Par exemple :
|               | ’TpSas ’C:/SAS/TpSas” |
| Username et Password | Informations d’authentification de l’utilisateur de base de données. |
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-in**: Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Table Name**

Renseignez le nom de la table à lire, préfixé par le nom du répertoire physique dans lequel elle est contenue. Par exemple : "TpSas.Customers".

**Query type**

La requête peut être **Built-in** ou distante dans le **Repository**

**Query**

Saisissez votre requête, si elle n’est pas stockée dans le **Repository**, en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.

**Advanced settings**

- **tStatCatcher Statistics**: Cochez cette case pour collecter les données de log au niveau du composant.
Global Variables


Utilisation

| Règle d’utilisation | Ce composant couvre toutes les possibilités de requête SQL dans les bases de données utilisant une connexion SAS. |

Scénarios associés

Pour un scénario associé, consultez :

- Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520.
tSasOutput

Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

⚠️ Avertissement :
Assurez-vous d’installer au préalable les modules sas.core.jar, sas.intrnet.javatools.jar et sas.svc.connection.jar sous le chemin `lib > java` dans votre build du Studio Talend pour pouvoir utiliser toutes les fonctionnalités des composants SAS. Vous pouvez vérifier que ces modules ont bien été installés à partir de la vue Modules de votre Studio Talend.

Le tSasOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

Propriétés du tSasOutput Standard

Ces propriétés sont utilisées pour configurer le tSasOutput s’exécutant dans le framework de Jobs Standard.

Le composant tSasOutput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

### Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

⚠️ Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

| SAS URL | Entrez le chemin d’accès à la base de données.

<p>| Driver JAR | Sélectionnez le driver JAR à partir de la liste déroulante ou cliquez sur les trois points [...] pour ajouter un nouveau JAR à la liste. |</p>
<table>
<thead>
<tr>
<th>Class Name</th>
<th>Dans le champ <strong>Class name</strong>, renseignez le nom de la classe pour qu'elle soit indiquée dans le driver.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username et Password</td>
<td>Informations d'authentification de l'utilisateur de base de données.  Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à lire.</td>
</tr>
</tbody>
</table>
| Action on data | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :  
**Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.  
**Update** : Mettre à jour les entrées existantes.  
**Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.  
**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.  
**Delete** : Supprimer les entrées correspondantes au flux d’entrée. |
<p>| Clear data in table | Cochez cette case pour que les données soient supprimées de la table sélectionnée avant toute action. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs. |</p>
<table>
<thead>
<tr>
<th>Built-In</th>
<th>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</th>
</tr>
</thead>
</table>
| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.  
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.  
Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com). |
| Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
- View schema : sélectionnez cette option afin de voir le schéma.  
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| Die on error | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décocchez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects. |

**Advanced settings**

<table>
<thead>
<tr>
<th>Commit every</th>
<th>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional Columns</td>
<td>Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</td>
</tr>
</tbody>
</table>
**Name** : Saisissez le nom de la colonne à modifier ou à insérer.

**SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.

**Position** : Sélectionnez **Before**, **Replace** ou **After**, en fonction de l'action à effectuer sur la colonne de référence.

**Reference column** : Saisissez une colonne de référence que le composant **tSasOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.

**Use field options** : Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.

**Enable debug mode** : Cochez cette case pour afficher chaque étape du processus d’écriture dans la base de données.

**tStatCatcher Statistics** : Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

- **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.
- **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable **After** et retourne un entier.
- **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable **After** et retourne un entier.
- **NB_LINE_DELETED** : nombre de lignes supprimées. Cette variable est une variable **After** et retourne un entier.
- **NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable **After** et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espase** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données utilisant une connexion SAS. Il permet de faire des actions sur une table ou les données d’une table d’une base de données SAS. Il permet aussi de créer un flux de rejet avec un lien Row &gt; Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.</th>
</tr>
</thead>
</table>

|-----------|--------------------------------------------------------------------------------------------------|

**Scénarios associés**

Pour un scénario associé au tSasOutput, consultez :

- Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667 du composant tMysqlOutput.
tSchemaComplianceCheck

Ce composant assure la qualité des données de n’importe quelle source de données en les comparant aux données de référence.

Le tSchemaComplianceCheck valide toutes les lignes d’entrée en fonction du schéma de référence ou vérifie les types, la nullabilité, la longueur de la ligne en fonction des valeurs de référence. Cette validation peut être appliquée à l’ensemble des lignes ou une partie seulement.
**tSCPClose**

Ce composant permet de fermer une connexion au protocole SCP. Le tSCPClose ferme la connexion au protocole entièrement crypté.

**Propriétés du tSCPClose Standard**

Ces propriétés sont utilisées pour configurer le tSCPClose s'exécutant dans le framework de Jobs Standard.

Le composant tSCPClose Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Component list | S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tSCPConnection dans la liste. |

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

**Global Variables**


**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie. |

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de |
contexte afin de sélectionner dynamiquement votre connexion SCP parmi celles prévues dans votre Job.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Ce composant est étroitement lié aux composants tSCPConnection et tSCPRollback. Il est généralement utilisé avec un composant tSCPConnection car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant tSCPClose, consultez Scénario : Gérer un fichier en utilisant SCP à la page 3699.
tSCPConnection

Ce composant ouvre une connexion SCP afin d’effectuer des transferts sécurisés de fichiers en une seule transaction.

Propriétés du tSCPConnection Standard

Ces propriétés sont utilisées pour configurer le tSCPConnection s’exécutant dans le framework de Jobs Standard.

Le composant tSCPConnection Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur SCP.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur SCP.</td>
</tr>
<tr>
<td>Username</td>
<td>Identifiant de connexion au serveur SCP.</td>
</tr>
<tr>
<td>Authentication method</td>
<td>Méthode d’authentification au serveur SCP.</td>
</tr>
<tr>
<td>Password</td>
<td>Mot de passe de connexion au serveur SCP.</td>
</tr>
</tbody>
</table>

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

Advanced settings

| tStatCatcher Statistics | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé comme sous-Job à un seul composant. |

**Scénarios associés**

Pour un scénario associé, consultez Scénario : Gérer un fichier en utilisant SCP à la page 3699.
**tSCPDelete**

Ce composant supprime un fichier d'un serveur SCP donné.

Le tSCPDelete supprime les fichiers sélectionnés d'un serveur distant par le biais d'un protocole entièrement crypté.

**Propriétés du tSCPDelete Standard**

Ces propriétés sont utilisées pour configurer le tSCPDelete s'exécutant dans le framework de Jobs Standard.

Le composant tSCPDelete Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur SCP.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur SCP.</td>
</tr>
<tr>
<td>Username</td>
<td>Identifiant de connexion au serveur SCP.</td>
</tr>
<tr>
<td>Authentication method</td>
<td>Méthode d'authentification au serveur SCP.</td>
</tr>
<tr>
<td>Password</td>
<td>Mot de passe de connexion au serveur SCP.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Filelist</td>
<td>Nom ou chemin d'accès du fichier à supprimer.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Statistique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher</td>
<td>Cochez la case afin de collecter les données de log au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td>NB_FILE</td>
<td>Indique le nombre de fichiers traités. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td>STATUS</td>
<td>résultats d'exécution du composant. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>
Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.

Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d'utilisation | Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé en association avec d'autres composants. |

**Scénario associé**

Aucun scénario n'est disponible pour la version Standard de ce composant.
tSCPFileExists

Ce composant vérifie la présence d'un fichier sur le serveur SCP spécifié.
Le tSCPFileExists vérifie si un fichier est présent sur le serveur distant, par le biais d'un protocole entièrement crypté.

**Propriétés du tSCPFileExists Standard**

Ces propriétés sont utilisées pour configurer le tSCPFileExists s'exécutant dans le framework de Jobs Standard.
Le composant tSCPFileExists Standard appartient à la famille Internet.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur SCP.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur SCP.</td>
</tr>
<tr>
<td>Username</td>
<td>Identifiant de connexion au serveur SCP.</td>
</tr>
<tr>
<td>Authentication method</td>
<td>Méthode d'authentification au serveur SCP.</td>
</tr>
<tr>
<td>Password</td>
<td>Mot de passe de connexion au serveur SCP. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Remote directory</td>
<td>Chemin d'accès au répertoire du serveur distant.</td>
</tr>
<tr>
<td>Filename</td>
<td>Nom du fichier à chercher.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

**Global Variables**

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaine de caracteres. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. FILENAME : nom du fichier traité. Cette variable est une variable Flow et retourne une chaine de caracteres. |
**STATUS** : résultats d'exécution du composant. Cette variable est une variable *After* et retourne une chaîne de caractères.

Une variable *Flow* fonctionne durant l'exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé en association avec d’autres composants. |

### Scénario : Gérer un fichier en utilisant SCP

Dans ce scénario, un Job de six composants vérifie qu’un fichier défini existe dans un répertoire spécifié sur un serveur SCP distant, puis retourne un message indiquant le résultat de cette vérification. Si le fichier n’existe pas dans ce répertoire, le Job met le fichier local sur le serveur SCP puis récupère le fichier depuis le serveur SCP vers un autre répertoire local défini afin de vérifier que le fichier a été correctement chargé.
Déposer et relier les composants

Procédure

1. Ajoutez les six composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette : tSCPConnection, tSCPFileExists, tJava, tSCPPut, tSCPGet et tSCPClose.
2. Reliez le tSCPConnection au tSCPFileExists à l’aide d’un lien Trigger > On Subjob Ok.
3. Reliez le tSCPFileExists au tJava à l’aide d’un lien Trigger > On Component Ok et au tSCPPut à l’aide d’un lien Trigger > Run if.
4. Reliez le tSCPPut au tSCPGet à l’aide d’un lien Trigger > On Subjob Ok.
5. Reliez le tSCPGet au tSCPClose à l’aide d’un lien Trigger > On Subjob Ok.

Configurer le composant tSCPConnection

Procédure

1. Double-cliquez sur le tSCPConnection afin d’ouvrir sa vue Basic settings.
2. Saisissez le nom d’hôte ou l’adresse IP du serveur SCP ainsi que le numéro de port dans les champs correspondants.
3. Saisissez le nom d’utilisateur utilisé pour cette connexion dans le champ **Username**.
4. Dans la liste **Authentication method**, sélectionnez **Password**.
5. Cliquez sur le bouton [...] à côté du champ **Password** puis, dans la boîte de dialogue [Enter a new password] qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** pour sauvegarder les changements.

**Configurer la vérification de l’existence du fichier**

**Procédure**

1. Double-cliquez sur le **tSCPFileExists** afin d’ouvrir sa vue **Basic settings**.

2. Cochez la case **Use an existing connection** puis, dans la liste **Component List**, sélectionnez la connexion que vous avez configurée.
3. Dans le champ **Filename**, saisissez le nom du fichier à vérifier entre guillemets doubles, "testscp.txt" dans cet exemple.
4. Double-cliquez sur le **tJava** afin d’ouvrir sa vue **Basic settings**.
5. Dans le champ **Code**, saisissez le code suivant :

```
System.out.println((String)globalMap.get("tSCPFileExists_1_FILENAME").
System.out.println((String)globalMap.get("tSCPFileExists_1_STATUS").
```

6. Sélectionnez l'onglet **Component** et cliquez sur le connexion **If** afin d'ouvrir sa vue **Basic settings**.

7. Dans le champ **Condition**, saisissez le code suivant :

```
!((String)globalMap.get("tSCPFileExists_1_STATUS").equals("File exists.")
```

Ainsi, si le fichier n'existe pas dans le répertoire défini, le reste du Job s'exécute.

**Configurer le composant tSCPPut**

**Procédure**

1. Double-cliquez sur le **tSCPPut** afin d'ouvrir sa vue **Basic settings**.

![tSCPPut](image)

2. Cochez la case **Use an existing connection** puis, dans la liste **Component List**, sélectionnez la connexion que vous avez configurée.

3. Cliquez sur le bouton [+ ] afin d'ajouter une ligne à la table **Filelist** puis saisissez le chemin complet vers le fichier local que vous souhaitez charger dans la colonne **Source** correspondante, "D:/SCPPut/testscp.txt" dans cet exemple.
Configurer le composant tSCPGet

Procédure
1. Double-cliquez sur le tSCPGet afin d’ouvrir sa vue Basic settings.

   ![Image of tSCPGet settings](image)

2. Cochez la case Use an existing connection puis, dans la liste Component List, sélectionnez la connexion que vous avez configurée.
3. Dans le champ Local directory, saisissez le chemin ou parcourez votre système vers le dossier dans lequel vous souhaitez stocker le fichier, "D:/SCPGet" dans cet exemple.
4. Dans la liste Overwrite or Append, sélectionnez append.
5. Cliquez sur le bouton [+] afin d’ajouter une ligne dans la table Filelist puis saisissez le chemin complet vers le fichier que vous souhaitez récupérer depuis le serveur SCP dans la colonne Source correspondante, "/testscp.txt" dans cet exemple.

Configurer le composant tSCPClose

Procédure
1. Double-cliquez sur le tSCPClose afin d’ouvrir sa vue Basic settings.

   ![Image of tSCPClose settings](image)

2. Dans la liste Component List, sélectionnez la connexion que vous avez configurée.

Exécuter le Job

Procédure
1. Appuyez sur Ctrl+S pour sauvegarder le Job.
2. Assurez-vous que le fichier testscp.txt existe dans le répertoire local D:/SCPPut et qu’il n’existe ni dans le répertoire local D:/SCPGet ni dans le répertoire racine du serveur SCP.
3. Appuyez sur F6 afin d’exécuter le Job.
Les informations concernant la vérification du fichier s’affichent dans la console Run.

4. Ouvrez le répertoire local D/SCPGet et le répertoire racine du serveur SCP afin de constater la présence du fichier testsscp.txt.
**tSCPFileList**

Ce composant liste les fichiers d'un serveur SCP spécifié.

Le tSCPFileList effectue une itération sur les fichiers d'un répertoire du serveur distant, par le biais d'un protocole entièrement crypté.

**Propriétés du tSCPFileList Standard**

Ces propriétés sont utilisées pour configurer le tSCPFileList s'exécutant dans le framework de Jobs Standard.

Le composant tSCPFileList Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th><strong>Use an existing connection</strong></th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat dans la liste Component list pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Spécifiez le nom d'hôte ou l'adresse IP du serveur SCP.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Spécifiez le numéro du port d'écoute du serveur SCP.</td>
</tr>
<tr>
<td><strong>Username</strong></td>
<td>Saisissez l'identifiant de connexion au serveur SCP.</td>
</tr>
<tr>
<td><strong>Authentication method</strong></td>
<td>Sélectionnez une des méthodes d'authentification au serveur SCP à partir de la liste déroulante : Public key, Password, ou Keyboard Interactive.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>Saisissez le mot de passe de connexion au serveur SCP.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td></td>
<td>Ce champ apparaît uniquement lorsque vous sélectionnez Password ou Keyboard Interactive dans la liste déroulante Authentication Method.</td>
</tr>
<tr>
<td><strong>Key Passphrase</strong></td>
<td>Saisissez la phrase secrète pour la clé publique.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir la phrase secrète, cliquez sur le bouton [...] à côté du champ Passphrase, puis, dans la boîte de dialogue qui s'ouvre, saisissez la phrase secrète entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td></td>
<td>Ce champ apparaît uniquement lorsque vous sélectionnez Public key dans la liste déroulante Authentication method.</td>
</tr>
<tr>
<td><strong>Private key</strong></td>
<td>Parcourez votre système ou saisissez le chemin d'accès à la clé privée. Ce champ apparaît uniquement lorsque vous sélectionnez <strong>Public key</strong> dans la liste déroulante <strong>Authentication method</strong>.</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Command Separator</strong></td>
<td>Spécifiez le caractère utilisé pour séparer les différentes commandes.</td>
</tr>
<tr>
<td><strong>Filelist</strong></td>
<td>Cliquez sur le bouton [+ ] pour ajouter une commande ou une série de commandes pour lister les fichiers d'un répertoire du serveur SCP.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| **tStatCatcher Statistics** | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

**Global Variables**

| **Global Variables** | **CURRENT_LINE** : fichier en cours d’itération. Cette variable est une variable **Flow** et retourne une chaîne de caractères.  
**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
**NB_LINE** : nombre de fichiers itérés jusqu'à présent. Cette variable est une variable **Flow** et retourne un entier.  
**STATUS** : résultats d’exécution du composant. Cette variable est une variable **After** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

**Utilisation**

| **Règle d’utilisation** | Ce composant peut être utilisé comme sous-Job à un seul composant ou en association avec d'autres composants.  
Ce composant est particulièrement pratique lorsqu’il est associé avec **tSCPGet** pour retrouver différents fichiers. |
Scénario associé

Pour un scénario associé utilisant un protocole différent, consultez Lister et obtenir des fichiers/dossiers d'un répertoire FTP à la page 1299.
tSCPGet

Ce composant permet de copier de fichiers du serveur SCP spécifié.

Le tSCPGet transfère les fichiers sélectionnés par le biais d'un protocole entièrement crypté via une connexion SCP.

**Propriétés du tSCPGet Standard**

Ces propriétés sont utilisées pour configurer le tSCPGet s’exécutant dans le framework de Jobs Standard.

Le composant tSCPGet Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur SCP.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d'écoute du serveur SCP.</td>
</tr>
<tr>
<td>Username</td>
<td>Identifiant de connexion au serveur SCP.</td>
</tr>
<tr>
<td>Authentication method</td>
<td>Méthode d'authentification au serveur SCP.</td>
</tr>
</tbody>
</table>
| Password | Mot de passe de connexion au serveur SCP.  
|          | Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| Local directory | Chemin d'accès du répertoire de destination. |
| Overwrite or Append | Liste des options disponibles pour le transfert des fichiers. |
| Filelist | Nom ou chemin d'accès du ou des fichier(s) à copier. |

**Advanced settings**

| tStatCatcher Statistics | Cochez la case afin de collecter les données de log au niveau de chaque composant. |

**Global Variables**

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
**NB_FILE** : Indique le nombre de fichiers traités. Cette variable est une variable After et retourne un entier.

**STATUS** : résultats d’exécution du composant. Cette variable est une variable After et retourne une chaîne de caractères.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme sous-job à un seul composant mais il peut aussi être utilisé en association avec d’autres composants. |

### Scénario associé

Pour un scénario associé, consultez Scénario : Gérer un fichier en utilisant SCP à la page 3699.
tSCPPut

Ce composant copie les fichiers dans le répertoire d’un serveur SCP distant.

Le tSCPPut copie les fichiers sélectionnés sur un serveur SCP distant par le biais d’un protocole entièrement crypté.

**Propriétés du tSCPPut Standard**

Ces propriétés sont utilisées pour configurer le tSCPPut s’exécutant dans le framework de Jobs Standard.

Le composant tSCPPut Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP d’un serveur SCP.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur SCP.</td>
</tr>
<tr>
<td><strong>Username</strong></td>
<td>Identifiant de connexion au serveur SCP.</td>
</tr>
<tr>
<td><strong>Authentication method</strong></td>
<td>Méthode d’authentification au serveur SCP.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>Mot de passe de connexion au serveur SCP.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Remote directory</strong></td>
<td>Chemin d’accès au répertoire de destination.</td>
</tr>
<tr>
<td><strong>Filelist</strong></td>
<td>Nom ou chemin d’accès au(x) fichier(s) à copier.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tStatCatcher</strong></td>
<td>Cochez la case afin de collecter les données de log au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td><strong>NB_FILE</strong></td>
<td>Indique le nombre de fichiers traités. Cette variable est une variable After et retourne un entier.</td>
</tr>
</tbody>
</table>
**STATUS** : résultats d’exécution du composant. Cette variable est une variable *After* et retourne une chaîne de caractères.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

---

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé en association avec d’autres composants. |

---

**Scénario associé**

Pour un scénario associé, consultez *Scénario : Gérer un fichier en utilisant SCP* à la page 3699.
tSCPRename

Ce composant renomme un ou des fichier(s) sur le serveur SCP sélectionné.

Propriétés du tSCPRename Standard

Ces propriétés sont utilisées pour configurer le tSCPRename s’exécutant dans le framework de Jobs Standard.

Le composant tSCPRename Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur SCP.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur SCP.</td>
</tr>
<tr>
<td>Username</td>
<td>Identifiant de connexion au serveur SCP.</td>
</tr>
<tr>
<td>Authentication method</td>
<td>Méthode d’authentification au serveur SCP.</td>
</tr>
<tr>
<td>Password</td>
<td>Mot de passe de connexion au serveur SCP.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>File to rename</td>
<td>Nom ou chemin d’accès au fichier à renommer.</td>
</tr>
<tr>
<td>Rename to</td>
<td>Saisissez le nouveau nom du fichier.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez la case afin de collecter les données de log au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Variables</td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option. <strong>STATUS</strong> : résultats d’exécution du composant. Cette variable est une variable After et retourne une chaîne de caractères. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est généralement utilisé comme sous-Job à un seul composant mais il peut aussi être utilisé en association avec d'autres composants.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Famille de composant</td>
<td>Internet/SCP</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé utilisant un protocole différent, consultez Renommer un fichier situé sur un serveur FTP à la page 1323.
tSCPTruncate

Ce composant supprime les données d’un ou plusieurs fichier(s) d’un serveur SCP via une connexion SCP.

Propriétés du tSCPTruncate Standard

Ces propriétés sont utilisées pour configurer le tSCPTruncate s’exécutant dans le framework de Jobs Standard.

Le composant tSCPTruncate Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur SCP.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur SCP.</td>
</tr>
<tr>
<td>Username</td>
<td>Identifiant de connexion au serveur SCP.</td>
</tr>
<tr>
<td>Authentication method</td>
<td>Méthode d’authentification au serveur SCP.</td>
</tr>
<tr>
<td>Password</td>
<td>Mot de passe de connexion au serveur SCP.</td>
</tr>
<tr>
<td>Remote directory</td>
<td>Chemin d’accès au fichier de destination.</td>
</tr>
<tr>
<td>Filelist</td>
<td>Nom ou chemin d’accès au(x) fichier(s) à vider.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher</td>
<td>Cochez la case afin de collecter les données de log au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td>NB_FILE</td>
<td>Indique le nombre de fichiers traités. Cette variable est une variable After et retourne un entier.</td>
</tr>
</tbody>
</table>
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé en tant que sous-Job à un seul composant mais il peut aussi être utilisé en association avec d’autres composants. |

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tSendMail**

Ce composant indique aux destinataires un état particulier d’un Job ou de possibles erreurs.
Le tSendMail envoie des messages et toute autre pièce jointe aux destinataires spécifiés.

**Propriétés du tSendMail Standard**

Ces propriétés sont utilisées pour configurer le tSendMail s'exécutant dans le framework de Jobs Standard.
Le composant tSendMail Standard appartient à la famille Internet.
Le composant de ce framework est toujours disponible.

**Basic settings**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>Adresse e-mail du destinataire principal.</td>
</tr>
<tr>
<td>From</td>
<td>Adresse e-mail du serveur d'envoi.</td>
</tr>
</tbody>
</table>
| Show sender’s name? | Cochez cette case pour définir le nom d’expéditeur à afficher dans les messages.  
  **Sendername** : renseignez le nom de l’expéditeur. |
| Cc | Adresse e-mail des destinataires en copie. |
| Bcc | Adresse e-mail des destinataires en copie cachée. |
| Subject | Objet du mail. |
| Message | Contenu du message. Appuyez sur **Ctrl+Espace** pour afficher la liste des variables disponibles. |
| Die if the attachment file doesn’t exist | Décochez cette case pour que les messages soient envoyés même si aucune pièce jointe n’est attachée. |

**Attachments**

Spécifiez les fichiers attachés et envoyés avec le message de l’e-mail. Cliquez sur le bouton [*] pour ajouter autant de lignes que nécessaire et configurez la valeur des colonnes suivantes pour chaque ligne :

- **File** : saisissez le chemin d'accès au fichier attaché ou le chemin d'accès au dossier si tous les fichiers du dossier sont en pièce jointe.
- **Content Transfer Encoding** : cliquez dans la cellule et sélectionnez dans la liste la valeur de l'encodage de transfert du contenu.

**Attachment / File and Content Transfer Encoding**

Cliquez sur le bouton [*] pour ajouter autant de lignes que nécessaire dans lesquelles vous pouvez ajouter des masques de fichier ou des chemins d'accès aux fichiers à envoyer dans l’e-mail, s'il y en a. Deux options sont disponibles pour l’encodage du contenu à transférer : **Default et Base64**.
**Other Headers**

Cliquez sur le bouton [+] pour ajouter des lignes de filtre :

**Key** : saisissez la clé de tout en-tête différent de l’en-tête standard.

**Value** : saisissez la valeur correspondante à la clé.

**SMTP Host et Port**

Adresse IP du serveur SMTP utilisé pour l’envoi d’emails.

**SSL Support**

Cochez cette case pour authentifier le serveur auprès du client via un protocole sécurisé SSL.

**STARTTLS Support**

Cochez cette case pour authentifier le serveur auprès du client via un protocole sécurisé STARTTLS.

**Importance**

 Sélectionnez dans la liste déroulante le niveau de priorité de votre message.

**Need authentication / Username and Password**

Cochez la case d’authentification et renseignez le nom de l’utilisateur (username) et son mot de passe (password), si cela est nécessaire pour accéder au service.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et les lignes contenant les erreurs seront ignorées.

---

**Advanced settings**

**MIME subtype from the ‘text’ MIME type**

 Sélectionnez dans la liste déroulante la forme à attribuer au texte du message.

**Encoding**

 Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

---

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaine de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé en tant que sous-Job mais peut aussi être utilisé comme composant de sortie ou de fin de Job. Il peut être connecté aux autres composants via des liens Row ou Iterate. |

### Scénario : Envoyer un e-mail lors d’une erreur

Ce scénario est un Job composé de trois composants permettant d’envoyer un e-mail à des destinataires spécifiques lorsqu’une erreur se produit.

- A partir de la **Palette**, cliquez-déposez les composants suivants dans l’espace de modélisation : **tFileInputDelimited**, **tFileOutputXML** et **tSendMail**.
- Cliquez-droit sur le composant **tFileInputDelimited** et reliez-le au composant **tFileOutputXML** à l’aide d’un lien de type **Row** > **Main**.
- Cliquez-droit sur le composant **tFileInputDelimited** et reliez-le au composant **tSendMail** à l’aide d’un lien de type **Trigger** > **OnSubjobError**.
- Double-cliquez sur le composant **tFileInputDelimited** pour afficher l’onglet **Basic settings** du composant et configurer ses paramètres de base.
A partir de la liste **Property type**, sélectionnez :

- **Repository** si vous avez déjà stocké les métadonnées de votre fichier d’entrée dans le **Repository**, les champs suivants sont alors renseignés automatiquement avec les informations stockées, ou

- **Built-in** si aucune métadonnée n’est stockée pour votre fichier d’entrée ; vous devez alors renseigner manuellement les champs suivants.

Dans cet exemple, c’est le mode **Built-in** qui est utilisé.

- Dans le champ **File Name**, renseignez le chemin d’accès ou cliquez sur le bouton [...] pour parcourir vos dossiers jusqu’au fichier à traiter. Ce champ est obligatoire.

Dans cet exemple, le fichier à traiter est **customer_mail.csv**.

- Définissez le séparateur de lignes dans le champ **Row separator** afin d’identifier la fin d’une ligne. Puis définissez le séparateur de champs dans **Field separator** pour délimiter les champs composant une ligne.

Les séparateurs utilisés dans ce scénario sont des séparateurs de champs et de lignes classiques de fichier CSV.

- Dans ce scénario, l’en-tête (**Header**), le pied de page (**Footer**) et la limite de lignes lues (**Limit**) n’ont pas besoin d’être définis.

- Sélectionnez l’option **Built-in** à partir de la liste déroulante **Schema** et renseignez manuellement les informations concernant le fichier délimité dans les champs correspondants.

- Cliquez sur le bouton [...] à côté du champ **Edit schema** pour afficher la boîte de dialogue permettant de définir la structure des données. Dans ce scénario, le schéma comporte quatre colonnes : **Id**, **Name**, **Address et IdState**.

- Cliquez sur le bouton [+] pour ajouter autant de lignes que nécessaire à la structure des données. Dans cet exemple, il n’y a qu’une seule colonne dans le schéma : **xmlStr**.

- Cliquez sur **OK** pour valider vos changements et fermer la boîte de dialogue.

**Remarque** :

Si vous avez préalablement stocké votre schéma dans le répertoire **Metadata** sous **File delimited**, sélectionnez l’option **Repository** dans la liste déroulante **Schema** puis cliquez sur le bouton [...] à côté du champ pour afficher la boîte de dialogue [**Repository Content**] à partir de laquelle vous pouvez sélectionner un schéma pertinent. Cliquez sur **OK** pour fermer la boîte de dialogue et remplir les champs automatiquement avec les métadonnées contenues dans le schéma.
Pour plus d’informations sur le stockage des schémas de métadonnées dans le Repository, consultez le Guide utilisateur du Studio Talend.

- Assurez-vous que la case **Skip empty rows** est cochée afin d’ignorer les lignes vierges.
- Double-cliquez sur le composant **tFileOutputXML** pour afficher l’onglet **Basic settings** du composant et configurer ses paramètres de base.

  ![tFileOutputXML](image)

  **Basic settings**
  - File Name: D:\Jobs\tFileinputXML\customer_data.xml
  - Row tag: "customer"
  - Schema: Built-In, Edit schema
  - Sync columns

- Cliquez sur le bouton [...] à côté du champ **File Name** pour sélectionner le répertoire de destination du fichier et saisissez le nom du fichier. Dans ce scénario, appelez le fichier *customer_data.xml*.
- Dans le champ **Row tag**, saisissez entre guillemets le nom que vous souhaitez donner à la balise contenant les données.
- Cliquez sur **Edit schema** et assurez-vous que le schéma est récupéré à partir du schéma d’entrée. Si nécessaire, cliquez sur **Sync columns** pour récupérer le schéma du composant précédent.
- Double-cliquez sur le composant **tSendMail** pour afficher l’onglet **Basic settings** du composant et configurer ses paramètres de base.
Saisissez les adresses e-mail des destinataires dans le champ **To** et les champs **Cc** et **Bcc** selon vos préférences, ainsi que celle de l’expéditeur dans le champ **From**.

Cochez la case **Show sender’s name?** puis renseignez le nom de l’expéditeur dans le champ **Sendername** pour que votre message ait une meilleure visibilité auprès de vos destinataires.

Renseignez l’objet du message dans le champ **Subject**.

Saisissez dans le champ **Message** le message contenant le code d’erreur produit à l’aide de la variable globale correspondante. Accédez à la liste des variables en appuyant sur **Ctrl+Espace**.

Ajoutez des pièces jointes et des informations sur les en-têtes, si nécessaire, dans les champs respectifs **Attachments** et **Other headers**.

Saisissez les informations SMTP dans les champs **SMTP host** et **SMTP port**.

Enregistrez votre Job puis appuyez sur **F6** pour l’exécuter.

Dans ce scénario, le fichier contenant les données à transférer vers une sortie XML ne peut être trouvé. Le composant **tSendMail** s’exécute lorsque cette erreur est rencontrée et envoie un mail de notification au destinataire spécifié.
**tServerAlive**

Ce composant valide le statut de la connexion à un hôte spécifié.

Le tServerAlive vérifie si la connexion à l’hôte défini est active ou non. Ceci est indiqué par la valeur booléenne retournée par la variable globale `SERVER_ALIVE_RESULT`.

**Propriétés du tServerAlive Standard**

Ces propriétés sont utilisées pour configurer le tServerAlive s’exécutant dans le framework de Jobs Standard.

Le composant tServerAlive Standard appartiennent à la famille Misc.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Protocol</strong></td>
<td>Dans la liste de protocoles, sélectionnez <strong>Ping</strong> ou <strong>Telnet</strong>.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP ou nom d’hôte du serveur distant à vérifier.</td>
</tr>
<tr>
<td><strong>Timeout Interval (in seconds)</strong></td>
<td>Définissez le délai avant suspension en secondes. Si l’hôte cible est atteint pendant l’intervalle défini, la valeur de la variable globale <code>SERVER_ALIVE_RESULT</code> est <em>true</em>. Sinon, sa valeur est <em>false</em>.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro de port d’écoute du serveur Telnet.</td>
</tr>
</tbody>
</table>

Cette option est disponible quand **Telnet** est sélectionné dans la liste **Protocol**.

**Advanced settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case afin de collecter les données de log au niveau du Job ainsi qu’au niveau des composants.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SERVER_ALIVE_RESULT</strong></td>
<td>statut de la connexion à un hôte distant. Cette variable est une variable <em>After</em> et retourne un booléen.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette
liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en tant que sous-job à un seul composant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td></td>
</tr>
</tbody>
</table>

### Scénario : Valider le statut de la connexion à un hôte distant

Dans ce scénario, le composant tServerAlive est utilisé afin d’envoyer une requête ping à un hôte distant. La valeur de sa variable globale SERVER_ALIVE_RESULT est extraite afin de connaître le statut de la connexion.

### Déposer et relier les composants

#### Procédure

1. Déposez les composants suivants depuis la Palette dans l’espace de modélisation graphique : un tServerAlive, un tFixedFlowInput et un tLogRow.
3. Reliez le tFixedFlowInput au tLogRow à l’aide d’un lien Row > Main.

![Diagramme de la configuration](image)

### Configurer les composants

#### Procédure

1. Double-cliquez sur le tServerAlive afin d’ouvrir sa vue Basic settings.
2. Dans la liste **Protocol**, sélectionnez **Ping**.


4. Dans le champ **Timeout Interval**, saisissez le délai avant suspension en secondes.

5. Double-cliquez sur le **tFixedFlowInput** afin d’ouvrir sa vue **Basic settings**.

6. Cliquez sur le bouton [...] à côté du champ **Edit schema** afin d’ouvrir l’éditeur de schéma.

7. Cliquez sur le bouton [+ ] afin d’ajouter une colonne de type booléen nommée **status**.

   Cliquez sur **OK** afin de valider les modifications et fermer l’éditeur.

   Dans la boîte de dialogue qui s’affiche, cliquez sur **Yes** afin de propager les changements aux autres composants.

8. Dans le champ **Value** de la table **Values**, appuyez sur **Ctrl + Espace** et sélectionnez **tServerAlive_1_SERVER_ALIVE_RESULT** dans la liste des variables.
9. Double-cliquez sur le tLogRow afin d'ouvrir sa vue Basic settings.

10. Sélectionnez Table (print values in cells of a table) pour un affichage optimal des résultats.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur Ctrl+S afin de sauvegarder votre Job.
2. Appuyez sur F6 afin d'exécuter le Job.

    [statistics] connecting to socket on port 3516
    [statistics] connected
    
    tLogRow_1
    status
    true

**Résultats**

Comme montré par la capture d'écran, le serveur distant est actif.
tServiceNowConnection

Ce composant ouvre une connexion à une instance de ServiceNow pouvant être réutilisée par d’autres composants ServiceNow.

Propriétés du tServiceNowConnection Standard

Ces propriétés sont utilisées pour configurer le tServiceNowConnection s’exécutant dans le framework de Jobs Standard.
Le composant tServiceNowConnection Standard appartient aux familles Business et Cloud.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>Saisissez l’URL de l’instance de ServiceNow à laquelle vous connecter.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d’authentification pour vous connecter à l’instance de ServiceNow. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi que des composants.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants ServiceNow. |

Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tServiceNowInput**

Ce composant accède à ServiceNow et en récupère des données.
Le tServiceNowInput récupère des données de ServiceNow selon les conditions spécifiées.

**Propriétés du tServiceNowInput Standard**

Ces propriétés sont utilisées pour configurer le tServiceNowInput s'exécutant dans le framework de Jobs Standard.
Le composant tServiceNowInput Standard appartient aux familles Business et Cloud.
Le composant de ce framework est toujours disponible.

**Basic settings**

| Use Existing Connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

**Remarque :**
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

| URL | Saisissez l'URL de l'instance de ServiceNow à laquelle vous vous connecter.

| Username et Password | Saisissez les informations d’authentification pour vous connecter à l’instance de ServiceNow.
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

| Table | Sélectionnez le nom de la table de laquelle récupérer les données ou sélectionnez Use Custom Table et, dans }
le champ **Custom Table Name** qui s’affiche, saisissez le nom de la table.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nomenclage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

### Conditions

Spécifiez les conditions de recherche pour récupérer uniquement les données correspondant aux critères en cliquant sur le bouton `[+]` pour ajouter autant de conditions que nécessaire. Configurez les paramètres suivants pour chaque condition :

- **Input column** : cliquez dans la cellule et sélectionnez la colonne du schéma d’entrée sur laquelle la condition de recherche doit être configurée.
- **Operator** : cliquez dans la cellule et sélectionnez un opérateur pour la condition de recherche.
- **Value** : saisissez la valeur pour la colonne correspondante pour la condition de recherche.

### Advanced settings

<table>
<thead>
<tr>
<th>Trim all columns</th>
<th>Cochez cette case pour supprimer les espaces en début et fin de toutes les colonnes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limit</td>
<td>Saisissez le nombre maximal d’enregistrements à récupérer.</td>
</tr>
</tbody>
</table>
### tStatCatcher Statistics

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début dans un flux Talend et nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.</td>
</tr>
<tr>
<td></td>
<td>Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les</td>
</tr>
</tbody>
</table>
Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tServiceNowOutput

Ce composant effectue une action définie sur les données de ServiceNow.
Le tServiceNowOutput insère ou met à jour des données dans ServiceNow.

Propriétés du tServiceNowOutput Standard

Ces propriétés sont utilisées pour configurer le tServiceNowOutput s'exécutant dans le framework de Jobs Standard.
Le composant tServiceNowOutput Standard appartient aux familles Business et Cloud.
Le composant de ce framework est toujours disponible.

Basic settings

| Use Existing Connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

| URL | Saisissez l’URL de l’instance de ServiceNow à laquelle vous connecter.

| Username et Password | Saisissez les informations d’authentification pour vous connecter à l’instance de ServiceNow.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvégarde les paramètres.

| Action | Sélectionnez dans la liste une action à effectuer sur les données :

- Insert : insérer les données dans ServiceNow.
### Table

 Sélectionnez le nom de la table à écrire ou sélectionnez **Use Custom Table** et, dans le champ **Custom Table Name** qui s’affiche, saisissez le nom de la table.

### Schema et Edit schema

 Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

 - **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


 Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

 - **View schema** : sélectionnez cette option afin de voir le schéma.

 - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

 - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Advanced settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable payload debug mode</td>
<td>Cochez cette case pour activer le mode <em>payload debug mode</em> et enregistrer le payload du message de requête.</td>
</tr>
<tr>
<td>Enable response debug mode</td>
<td>Cochez cette case pour activer le mode <em>response debug mode</em> et enregistrer le message de réponse du serveur de logs.</td>
</tr>
</tbody>
</table>

### tStatCatcher Statistics

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi que des composants.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Variables</td>
<td><strong>NB_LINE</strong> : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
</tbody>
</table>
**NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.

**NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.

**NB_LINE_REJECTED** : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.

**PAYLOAD** : payload du message de requête JSON. Cette variable est une variable Flow et retourne un objet.

**RESPONSE** : message de réponse du serveur de logs. Cette variable est une variable Flow et retourne un objet.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de fin dans un flux Talend et nécessite un composant d’entrée.</th>
</tr>
</thead>
</table>

| Dynamic settings    | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions.
Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tSetEnv

Ce composant ajoute des variables à l’environnement du système de manière temporaire, pendant l’exécution d’un Job.

Le tSetEnv permet de créer des variables et d’exécuter un Job script en communicant les informations des variables nouvellement créées entre les différents sous-Jobs. Après l’exécution du Job, ces nouvelles variables sont supprimées.

Propriétés du tSetEnv Standard

Ces propriétés sont utilisées pour configurer le tSetEnv s’exécutant dans le framework de Jobs Standard.

Le composant tSetEnv Standard appartient à la famille System.

Le composant de ce framework est toujours disponible.

Basic settings

| Parameters | Cliquez sur le bouton [+] pour ajouter les variables nécessaires au Job. name : Renseignez la syntaxe de la nouvelle variable. value : Entrez une valeur pour ce paramètre en fonction du contexte. append : Cochez cette case pour ajouter la nouvelle variable à la fin. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables


Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
Utilisation

| Règle d’utilisation | Le tSetEnv peut être utilisé comme composant de début ou composant intermédiaire. |

**Scénario : Modifier une variable lors de l’exécution d’un Job**

Le scénario suivant se présente sous la forme de deux Jobs père et fils. Il permet grâce au tSetEnv de transférer et de modifier dans un Job fils une valeur créée dans un Job parent. Dans le cadre de ce Job, les composants tMsgBox sont utilisés pour afficher à titre informatif qu’une variable est correctement appliquée, via une fenêtre pop-up.

Pour modifier la valeur du Job père en utilisant une variable créée par le tSetEnv dans le Job fils, procédez comme décrit dans les sections suivantes :

**Déposer et relier les composants**

**Procédure**

2. A partir de la Palette, glissez-déposez un composant tSetEnv, deux composants tMsgBox et un tRunJob dans l’espace de modélisation graphique.
3. Connectez le tSetEnv à un premier tMsgBox à l’aide d’un lien OnSubjobOk : cliquez-droit sur le composant de départ, sélectionnez Trigger, puis OnSubjobOk, puis cliquez sur le composant à relier.
4. Connectez le premier tMsgBox au tRunJob également à l’aide d’un lien OnSubjobOk.
5. Connectez ensuite le tRunJob au second tMsgBox également à l’aide d’un lien OnSubjobOk.

7. A partir de la Palette, glissez-déposez un tSetEnv et un tMsgBox.
8. Connectez le tSetEnv au tMsgBox à l’aide d’un lien OnSubjobOk : cliquez-droit sur le composant de départ, sélectionnez Trigger, puis OnSubjobOk. Puis cliquez sur le composant à relier.
Configurer les composants

Pourquoi et quand exécuter cette tâche
Dans cet exemple, la valeur définie dans le Job parent est transférée au Job fils. Elle y est modifiée pour prendre la valeur du Job fils et retransférée vers le Job parent.

Procédure
1. Dans le Job ParentJob, sélectionnez le tSetEnv puis cliquez sur l’onglet Component. Ajoutez une ligne de variable en cliquant sur le bouton [+] pour définir sa valeur initiale. Saisissez Variable_1 dans le champ Name et Parent job value dans le champ Value.
2. Sélectionnez le premier tMsgBox et cliquez sur l’onglet Component. Dans le champ Message, saisissez le message qui s’affiche dans votre fenêtre pop-up et qui confirme que votre variable est bien prise en compte dans le Job. Par exemple: "Parent:"+System.getProperty("Variable_1") affiche la variable définie dans le tSetEnv (ici Parent job value).
3. Sélectionnez le second tMsgBox et cliquez sur l’onglet Component. Dans le champ Message, saisissez à nouveau la ligne "Parent:"+System.getProperty("Variable_1"). Elle fait apparaître dans le Job parent la variable définie dans le Job fils.

5. Double-cliquez sur le tRunJob pour ouvrir le Job fils ChildJob.
6. Sélectionnez le composant **tSetEnv** et cliquez sur l’onglet **Component**. Ajoutez une ligne de variable en cliquant sur le bouton [+] pour définir sa valeur initiale. Saisissez **Variable_1** dans le champ **Name** et **Child job value** dans le champ **Value**.

7. Sélectionnez ensuite le **tMsgBox** et cliquez sur l’onglet **Component**. Dans le champ **Message**, saisissez le message qui s’affiche dans votre fenêtre pop-up et qui confirme que votre variable est correctement prise en compte dans le Job. Par exemple, la ligne de code "Son:” + System.getProperty("Variable_1") affiche la variable définie dans le **tSetEnv** (ici **Child job value**).

<table>
<thead>
<tr>
<th>Title</th>
<th>“Talend Open Studio”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buttons</td>
<td>OK</td>
</tr>
<tr>
<td>Icon</td>
<td>Icon Information</td>
</tr>
<tr>
<td>Message</td>
<td>“Son:” + System.getProperty(&quot;Variable_1&quot;)</td>
</tr>
</tbody>
</table>

8. Enregistrez votre Job, retournez sous le Job parent, puis appuyez sur **F6** pour l’exécuter.

**Exécuter le Job**

Trois fenêtres pop-up s’affichent l’une après l’autre :

- **Parent: Parent job value** : la valeur du Job parent est **Parent job value**.
- **Child: Child job value** : la valeur du Job fils est **Child job value**.
- **Parent: Child job value** : la valeur du Job parent a été modifiée par la variable définie dans le **tSetEnv** du Job fils, puis retransférée au Job parent. La valeur du Job parent est dorénavant celle définie dans le Job fils : **Child job value**.
tSetGlobalVar

Ce composant simplifie le processus de définition des variables globales.
Le tSetGlobalVar vous permet de définir et de configurer les variables globales dans l’interface graphique.

Propriétés du tSetGlobalVar Standard

Ces propriétés sont utilisées pour configurer le tSetGlobalVar s’exécutant dans le framework de Jobs Standard.
Le composant tSetGlobalVar Standard appartient à la famille Custom Code.
Le composant de ce framework est toujours disponible.

Basic settings

| Variables | Ce tableau contient deux colonnes. Key : Nom de la variable à appeler dans le code. Value : la valeur assignée à cette variable. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

Global Variables


Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme un sous-job à un composant. |
Scénario : Afficher le contenu d'une variable globale

Ce scénario se compose d'un Job simple affichant la valeur d'une variable globale définie dans le composant tSetGlobalVar.

1. Déposez les composants suivants de la Palette dans l'espace de modélisation graphique : tSetGlobalVar et tJava.
2. Reliez le tSetGlobalVar au tJava à l'aide d'un lien Trigger > OnSubjobOk.

Configurer le composant tSetGlobalVar

1. Double-cliquez sur le composant tSetGlobalVar afin d'afficher sa vue Basic settings.
2. Cliquez sur le bouton [+] pour ajouter une ligne dans la table Variables, puis renseignez les colonnes Key et Value en saisissant respectivement K1 et 20.
3. Double-cliquez ensuite sur le composant tJava afin d'afficher sa vue Basic settings.
4. Dans la zone **Code**, saisissez le code suivant :

```java
String foo = "bar";
String K1;
String Result = "The value is:";
Result = Result + globalMap.get("K1");
System.out.println(Result);
```

Dans ce scénario, utilisez la variable **Result**. Pour accéder à la liste des variables globales, appuyez sur les touches **Ctrl + Espace** de votre clavier et sélectionnez les paramètres correspondants.

**Exécuter le Job**

Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

```
Starting job SetGlobalVar at 16:22 25/02/2011
[statistics] connecting to socket on port 3489
[statistics] connected
The value is:20
[statistics] disconnected
Job SetGlobalVar ended at 16:22 25/02/2011. [exit code=0]
```

Le contenu de la variable globale **K1** s’affiche dans la console.
tSetKerberosConfiguration

Ce composant configure informations d'authentification via Kerberos.

Le tSetKerberosConfiguration est conçu pour configurer l'authentification via Kerberos afin d'améliorer la sécurité des communications réseau.

Pour plus d'informations concernant le protocole Kerberos, consultez http://www.kerberos.org (en anglais).

Propriétés du tSetKerberosConfiguration Standard

Ces propriétés sont utilisées pour configurer le tSetKerberosConfiguration s'exécutant dans le framework de Jobs Standard.

Le composant tSetKerberosConfiguration Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDC Server</td>
<td>Adresse du centre de distribution de clés (Key Distribution Center, KDC).</td>
</tr>
<tr>
<td>Realm</td>
<td>Nom du royaume (realm) de Kerberos.</td>
</tr>
<tr>
<td>Username and Password</td>
<td>Informations d'authentification de Kerberos. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour rassembler les métadonnées de traitement du Job au niveau du Job et au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td>Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant.</td>
<td></td>
</tr>
<tr>
<td>Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette</td>
<td></td>
</tr>
</tbody>
</table>
liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en tant que sous-job et est utilisé avec le tSoap.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Le composant tSetKerberosConfiguration ne fonctionne pas avec une JVM IBM.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tSetKeystore**

Ce composant définit le type des données d'authentification entre **PKCS 12** et **JKS**.

Le tSetKeystore soumet des données d'authentification d'un TrustStore avec ou sans Keystore pour valider la connexion SSL.

**Propriétés du tSetKeystore Standard**

Ces propriétés sont utilisées pour configurer le tSetKeystore s'exécutant dans le framework de Jobs Standard.

Le composant tSetKeystore Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TrustStore type</strong></td>
<td>Sélectionnez le type TrustStore à utiliser. Il peut être <strong>PKCS 12</strong> ou <strong>JKS</strong>.</td>
</tr>
<tr>
<td><strong>TrustStore file</strong></td>
<td>Saisissez le chemin d'accès et le nom du fichier TrustStore, ou parcourez votre répertoire jusqu'au TrustStore contenant la liste des certificats approuvés par le client.</td>
</tr>
<tr>
<td><strong>TrustStore password</strong></td>
<td>Saisissez le mot de passe utilisé pour vérifier l’intégrité des données TrustStore.</td>
</tr>
<tr>
<td>Need Client authentication</td>
<td>Cochez cette case pour valider les données keystore. Vous devez ensuite compléter trois champs :</td>
</tr>
<tr>
<td></td>
<td>- <strong>KeyStore type</strong> : sélectionnez le type de keystore à utiliser. Il peut être <strong>PKCS 12</strong> ou <strong>JKS</strong>.</td>
</tr>
<tr>
<td></td>
<td>- <strong>KeyStore file</strong> : saisissez le chemin d'accès et le nom du fichier Keystore, ou parcourez votre répertoire jusqu'au fichier contenant les données keystore.</td>
</tr>
<tr>
<td></td>
<td>- <strong>KeyStore password</strong> : saisissez le mot de passe keystore.</td>
</tr>
<tr>
<td><strong>Check server identity</strong></td>
<td>Cochez cette case afin que le Job vérifie la correspondance entre le nom de l’hôte de l’URL et le nom de l’hôte du serveur. S’ils ne correspondent pas, le mécanisme de vérification vous demande si la connexion doit être autorisée.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu'au niveau de chaque composant.</td>
</tr>
</tbody>
</table>
Global Variables


Utilisation

| Règle d’utilisation | Ce composant est utilisé en standalone. |
| Connections | Liens de sortie (de ce composant à un autre) :


Liens d’entrée (d’un autre composant à celui-ci) :


Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend. |

Scénario : Extraire des informations client d’un fichier WSDL privé

Ce scénario décrit un Job comprenant trois composants, qui se connecte à un fichier WSDL privé, pour extraire des informations client.

Le fichier WSDL utilisé dans ce Job accède au service Web correspondant via le protocole SSL. Le code correspondant dans le fichier est le suivant :

```xml
<wsdl:port name="CustomerServiceHttpSoap11Endpoint"
  binding="ns:CustomerServiceSoap11Binding">
  <soap:address location="https://192.168.0.22:8443/axis2/services/
  CustomerService.CustomerServiceHttpSoap11Endpoint/"/>
</wsdl:port>
```

Saisissez le code suivant dans le fichier server.xml de Tomcat :

```xml
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
  maxThreads="150" scheme="https" secure="true"
  clientAuth="true" sslProtocol="TLS"/>```
Vous avez besoin de fichiers Keystore pour vous connecter à ce fichier WSDL. Pour reproduire ce Job, procédez comme suit :

- Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : `tSetKeystore`, `tWebService`, et `tLogRow`.

  ![Diagramme de processus](image)

- Cliquez-droit sur le composant `tSetKeystore` pour ouvrir son menu contextuel.
- Dans ce menu, sélectionnez `Trigger > On Subjob Ok` afin de relier ce composant au `tWebService`.
- Cliquez-droit sur le `tWebService` pour ouvrir son menu contextuel.
- Dans ce menu, sélectionnez `Row > Main` pour connecter ce composant au `tLogRow`.
- Double-cliquez sur le `tSetKeystore` pour ouvrir sa vue `Basic settings` et définir ses propriétés.

  ![Vue Basic settings](image)

- Dans la liste `TrustStore type`, sélectionnez `PKCS12`.
- Dans le champ `TrustStore file`, parcourez votre répertoire jusqu’au fichier TrustStore. Ici, le fichier est `server.p12`.
- Dans le champ `TrustStore password`, saisissez le mot de passe pour ce fichier TrustStore. Dans cet exemple, le mot de passe est `password`.
- Cochez la case `Need Client authentication` afin d’activer les champs de configuration Keystore.
• Dans la liste **KeyStore type**, sélectionnez **JKS**.

• Dans le champ **KeyStore file**, parcourez votre répertoire jusqu’au fichier Keystore correspondant. Ici, le fichier est **server.keystore**.

• Double-cliquez sur le composant **tWebService** pour ouvrir l’éditeur du composant, ou sélectionnez le composant dans l’espace de modélisation graphique et dans sa vue **Basic settings**, cliquez sur le bouton [...] à côté du champ **Service configuration**.

![WSDL Input mapping Output mapping]

- Dans le champ **WSDL**, parcourez votre répertoire jusqu’au fichier WSDL privé à utiliser. Dans cet exemple, ce fichier est **CustomerService.wsdl**.

- Cliquez sur le bouton **Refresh** à côté du champ **WSDL** afin de récupérer la description WSDL et l’afficher dans les champs qui suivent.

- Dans la liste **Port Name**, sélectionnez le port que vous souhaitez utiliser, **CustomerServiceHttpSoap11Endpoint** dans cet exemple.

- Dans la liste **Operation**, sélectionnez le service que vous souhaitez utiliser. Dans cet exemple, le service sélectionné est **getCustomer(parameters):Customer**.

- Cliquez sur **Next** pour ouvrir une nouvelle vue dans l’éditeur.
Dans le panneau de droite de la vue **Input mapping**, le paramètre d’entrée du service s’affiche automatiquement. Cependant, vous pouvez ajouter d’autres paramètres si vous sélectionnez [+] параметры et que vous cliquez sur le bouton [+] en haut, afin d’afficher la boîte de dialogue [Parameter Tree] dans laquelle vous pouvez sélectionner n’importe lequel des paramètres listés.

Le service Web de cet exemple n’a qu’un seul paramètre d’entrée, **ID**.

- Dans la colonne **Expression** de la ligne **parameters.ID**, saisissez entre guillemets l’ID du client qui vous intéresse. Dans cet exemple, l’ID est **A00001**.

- Cliquez sur **Next** pour ouvrir une nouvelle vue dans l’éditeur.

Dans la liste **Element** à gauche de la vue s’affiche automatiquement le paramètre de sortie du service Web. Cependant, vous pouvez ajouter d’autres paramètres si vous sélectionnez [+] parameters et que vous cliquez sur le bouton [+] en haut afin d’afficher la boîte de dialogue [Parameter Tree] dans laquelle vous pouvez sélectionner n’importe lequel des paramètres affichés.

Le service Web dans cet exemple contient quatre paramètres de sortie : **return.address**, **return.email**, **return.name** et **return.phone**.

Vous devez créer une connexion entre le paramètre de sortie du service Web défini et le schéma du composant de sortie. Pour ce faire :

```plaintext
parameters.ID "A00001" "parameters.ID
```

Dans la colonne **Expression** de la ligne **parameters.ID**, saisissez entre guillemets l’ID du client qui vous intéresse. Dans cet exemple, l’ID est **A00001**.
Dans le panneau à droite de la vue, cliquez sur le bouton [...] à côté du champ **Edit Schema** afin d’ouvrir une boîte de dialogue dans laquelle vous pouvez définir le schéma de sortie.

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Type</th>
<th>✓</th>
<th>N.</th>
<th>Date Pattr...</th>
<th>Length</th>
<th>Pre...</th>
<th>D...</th>
<th>Co...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
<td>String</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phone</td>
<td></td>
<td>String</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td></td>
<td>String</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td></td>
<td>String</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dans l’éditeur de schéma, cliquez sur le bouton [+] pour ajouter quatre colonnes au schéma de sortie.

Cliquez dans chaque colonne et saisissez les nouveaux noms : **Name, Phone, Email et Address** dans cet exemple. Cela vous permettra de récupérer les informations client qui vous intéressent.

Cliquez sur **OK** pour valider les modifications et fermer la boîte de dialogue du schéma.

Dans la liste **Element** à droite de l’éditeur, glissez chaque paramètre dans le champ correspondant à la colonne que vous avez définie dans l’éditeur de schéma.

**Remarque :**

S’il est disponible, vous pouvez utiliser le bouton **Auto map!**, situé en bas à gauche de l’interface, afin d’effectuer l’opération de mapping automatiquement.

Cliquez sur **OK** pour valider vos modifications et fermer l’éditeur.

Dans l’espace de modélisation graphique, double-cliquez sur le composant **tLogRow** afin d’ouvrir sa vue **Basic settings** et définir ses propriétés.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

Sauvegardez votre Job et appuyez sur **F6** pour l’exécuter.

Les informations du client dont l’ID est **A00001** sont retournées et affichées dans la console du **Studio Talend**.

```
<table>
<thead>
<tr>
<th>name</th>
<th>phone</th>
<th>email</th>
<th>address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rose Gonzalez</td>
<td>(512) 757-9000</td>
<td><a href="mailto:rose@edge.com">rose@edge.com</a></td>
<td>313 Constitution Place Austin, TX 78767 USA</td>
</tr>
</tbody>
</table>
```

Job tSetKeystore ended at 19:01 24/11/2010. (exit code=0)
tSetProxy

Ce composant configure les informations de configuration de votre proxy.

Propriétés du tSetProxy Standard

Ces propriétés sont utilisées pour configurer le tSetProxy s’exécutant dans le framework de Jobs Standard.

Le composant tSetProxy Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Basic settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy type</td>
<td>Liste des protocoles de proxy.</td>
</tr>
<tr>
<td>Proxy host</td>
<td>Adresse du serveur du proxy.</td>
</tr>
<tr>
<td>Proxy port</td>
<td>Numéro du port du proxy.</td>
</tr>
<tr>
<td>Proxy user</td>
<td>Identifiant d’authentification pour le proxy.</td>
</tr>
<tr>
<td>Proxy password</td>
<td>Mot de passe d’authentification pour le proxy.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>None proxy hosts</td>
<td>Les hôtes auxquels vous devez vous connecter directement et non via le serveur de proxy. La valeur peut être une liste d’hôtes, séparés par le caractère</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Advanced settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

**Utilisation**

| Règle d'utilisation | Le **tSetProxy** peut être utilisé en sous-job indépendant, mais peut également être déployé avec d'autres composants de la famille Internet. |

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tSleep**

Ce composant identifie les goulots d’étranglement éventuels à l’aide d’une pause lors de l’exécution du Job à des fins de test ou de monitoring.

Le composant tSleep effectue une pause lors de l’exécution d’un Job. En production, il peut être utilisé pour tout type de pause dans le traitement d’un Job, pour alimenter un flux entrant par exemple.

**Propriétés du tSleep Standard**

Ces propriétés sont utilisées pour configurer le tSleep s’exécutant dans le framework de Jobs Standard.

Le composant tSleep Standard appartient à la famille Orchestration.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Pause (in second)</th>
<th>Temps en secondes pendant lequel le Job est suspendu.</th>
</tr>
</thead>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</th>
</tr>
</thead>
</table>

**Global Variables**

|-------------------|----------------------------------------------------------------------------------------------------------------------------------|

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Le composant tSleep est généralement utilisé comme composant intermédiaire mettant un Job en pause, avant de le reprendre.</th>
</tr>
</thead>
</table>
### tSleep

| Connections | Liens de sortie (de ce composant à un autre) :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Row</strong> : Main, Iterate.</td>
</tr>
<tr>
<td></td>
<td><strong>Trigger</strong> : OnSubjobOk, OnSubjobError, Run if,</td>
</tr>
<tr>
<td></td>
<td>OnComponentOk, OnComponentError.</td>
</tr>
</tbody>
</table>

|             | Liens d’entrée (d’un autre composant à celui-ci) :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Row</strong> : Main, Reject, Iterate.</td>
</tr>
<tr>
<td></td>
<td><strong>Trigger</strong> : OnSubjobOk, OnSubjobError, Run if,</td>
</tr>
<tr>
<td></td>
<td>OnComponentOk, OnComponentError, Synchronize,</td>
</tr>
<tr>
<td></td>
<td>Parallelize.</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.

### Scénario associé

Pour un exemple d’utilisation du **tSleep**, consultez *Procédure* à la page 2110.
**tSnowflakeClose**

Ce composant ferme une connexion active à Snowflake afin de libérer les ressources occupées.

**Propriétés du tSnowflakeClose Standard**

Ces propriétés sont utilisées pour configurer le tSnowflakeClose s’exécutant dans le framework de Jobs Standard.

Le composant tSnowflakeClose Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

<i>Remarque :</i>

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Component</td>
<td>Sélectionnez dans la liste déroulante le composant établissant la connexion à fermer.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Variables globales**

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Snowflake, particulièrement avec le tSnowflakeConnection. |
Scénario associé

Pour un scénario associé, consultez Écrire et lire des données dans une table Snowflake à la page 3763.
tSnowflakeConnection

Ce composant ouvre une connexion à Snowflake, pouvant être réutilisée par d’autres composants Snowflake.

Propriétés du tSnowflakeConnection Standard

Ces propriétés sont utilisées pour configurer le tSnowflakeConnection s’exécutant dans le framework de Jobs Standard.

Le composant tSnowflakeConnection Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• Built-In : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• Repository : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td>Account</td>
<td>Dans le champ Account, saisissez, entre guillemets doubles, le nom qui vous a été assigné par Snowflake.</td>
</tr>
<tr>
<td>User Id et Password</td>
<td>Saisissez entre guillemets doubles, vos informations d’authentification à Snowflake.</td>
</tr>
<tr>
<td></td>
<td>• Dans le champ User ID, saisissez, entre guillemets doubles, votre identifiant défini dans Snowflake via le paramètre LOGIN_NAME de Snowflake. Pour plus d’informations, contactez l’administrateur de votre système Snowflake.</td>
</tr>
<tr>
<td></td>
<td>• Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte</td>
</tr>
</tbody>
</table>
de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

<table>
<thead>
<tr>
<th>Warehouse</th>
<th>Saisissez, entre guillemets doubles, le nom de l’entrepôt Snowflake à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Saisissez, entre guillemets doubles, le nom du schéma de la base de données à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez, entre guillemets doubles, le nom de la base de données Snowflake à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Login Timeout</th>
<th>Spécifiez le temps d’attente d’une réponse lors de la connexion à Snowflake, avant de retourner une erreur.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracing</td>
<td>Sélectionnez le niveau de log pour le pilote JDBC de Snowflake. Si cette option est activée, un log standard Java est généré.</td>
</tr>
<tr>
<td>Role</td>
<td>Saisissez, entre guillemets doubles, le rôle de contrôle des accès par défaut à utiliser pour initialiser la session Snowflake. Ce rôle doit déjà exister et doit avoir été assigné à l’ID de l’utilisateur que vous utilisez pour vous connecter Snowflake. Si vous laissez ce champ vide, le rôle PUBLIC est automatiquement assigné. Pour plus d’informations concernant le modèle de contrôle des accès de Snowflake, consultez la documentation Snowflake à l’adresse suivante Understanding the Access Control Model (en anglais).</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Variables globales**

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

**Utilisation**

| Règle d’utilisation | Ce composant s’utilise généralement avec d’autres composants Snowflake. Il crée une connexion à Snowflake et fonctionne avec d’autres sous-jobs réutilisant cette connexion, à l’aide d’un lien Trigger. |
Scénario associé

Pour un scénario associé, consultez Écrire et lire des données dans une table Snowflake à la page 3763.
tSnowflakeInput

Ce composant lit des données d'une table Snowflake dans le flux de données de votre Job selon une requête SQL.

**Propriétés du tSnowflakeInput Standard**

Ces propriétés sont utilisées pour configurer le tSnowflakeInput s'exécutant dans le framework de Jobs Standard.

Le composant tSnowflakeInput Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d'un connecteur à une base de données dynaïmique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton […] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
<tr>
<td>Connection Component</td>
<td>Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant.</td>
</tr>
<tr>
<td>Account</td>
<td>Dans le champ Account, saisissez, entre guillemets doubles, le nom qui vous a été assigné par Snowlake.</td>
</tr>
<tr>
<td>User Id et Password</td>
<td>Saisissez entre guillemets doubles, vos informations d’authentification à Snowflake.</td>
</tr>
</tbody>
</table>
Dans le champ **User ID**, saisissez, entre guillemets doubles, votre identifiant défini dans Snowflake via le paramètre `LOGIN_NAME` de Snowflake. Pour plus d’informations, contactez l’administrateur de votre système Snowflake.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

<table>
<thead>
<tr>
<th>Warehouse</th>
<th>Saisissez, entre guillemets doubles, le nom de l’entrepôt Snowflake à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Saisissez, entre guillemets doubles, le nom du schéma de la base de données à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez, entre guillemets doubles, le nom de la base de données Snowflake à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</td>
</tr>
<tr>
<td>Table</td>
<td>Cliquez sur le bouton [...] et, dans l’assistant qui s’affiche, sélectionnez la table Snowflake à utiliser.</td>
</tr>
</tbody>
</table>
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.  
**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
Si le type de données Snowflake à gérer est VARIANT, OBJECT ou ARRAY, lorsque vous définissez le schéma dans le composant, sélectionnez *String* pour les données correspondantes dans la colonne **Type** de l’éditeur de schéma.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• **View schema** : sélectionnez cette option afin de voir le schéma.  
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications. |
modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Notez que, si la valeur d’entrée de tout champ primitif non nullable est nulle, la ligne de données comprenant ce champ sera rejetée.

**Manual query**

Cochez cette case et, dans le champ **Full SQL query string**, saisissez, entre guillemets doubles, l’instruction de requête à exécuter. Pour plus d’informations relatives aux commandes de Snowflake SQL, consultez SQL Command Reference (en anglais).

**Condition**

Saisissez, entre guillemets doubles, l’expression booléenne à utiliser pour filtrer les données de votre table Snowflake. Cette expression est utilisée comme une clause WHERE dans la requête SELECT exécutée par le tSnowflakeInput.

### Advanced settings

**Login Timeout**

Spécifiez le temps d’attente d’une réponse lors de la connexion à Snowflake, avant de retourner une erreur.

**Tracing**

Sélectionnez le niveau de log pour le pilote JDBC de Snowflake. Si cette option est activée, un log standard Java est généré.

**Role**

Saisissez, entre guillemets doubles, le rôle de contrôle des accès par défaut à utiliser pour initialiser la session Snowflake.

Ce rôle doit déjà exister et doit avoir été assigné à l’ID de l’utilisateur que vous utilisez pour vous connecter Snowflake. Si vous laissez ce champ vide, le rôle PUBLIC est automatiquement assigné. Pour plus d’informations concernant le modèle de contrôle des accès de Snowflake, consultez la documentation Snowflake à l’adresse suivante Understanding the Access Control Model (en anglais).

**Convert columns and table to uppercase**

Cochez cette case pour passer les minuscules en majuscules, dans le nom de la table et des colonnes du schéma définies.

Cette propriété n’est pas disponible lorsque la case Manual Query est cochée.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.
Variables globales

<table>
<thead>
<tr>
<th>Variable</th>
<th>Définition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Utilisation

| Règle d’utilisation | Ce composant est un composant de début de flux de données dans votre Job. Il envoie des données à d’autres composants via un lien Row. |

Écrire et lire des données dans une table Snowflake

Ce scénario fournit un exemple d’utilisation des composants Talend Snowflake pour vous connecter à Snowflake, créer une table Snowflake, écrire des données dans la table, lire des données de cette table, puis fermer la connexion à Snowflake.
Créer un Job écrivant et lisant des données Snowflake

Créez un Job pour vous connecter à Snowflake, créer une table Snowflake, écrire des données dans la table, lire les données de cette table puis fermer la connexion à Snowflake.

Avant de commencer

- L’entrepôt de données Snowflake à utiliser doit être démarré et avoir un statut différent de Suspended.
- Le rôle Snowflake qui vous est assigné doit comporter des droits en lecture écriture sur la base de données à laquelle vous allez vous connecter.

Si vous n’êtes pas sûr de respecter ces prérequis, contactez l’administrateur de votre système Snowflake.

Procédure

1. Dans la perspective Integration du Studio, créez un nouveau Job à partir du nœud Job Designs de la vue Repository.
2. Ajoutez un composant tSnowflakeConnection, un tSnowflakeRow, un tFixedFlowInput, un tSnowflakeOutput, un tSnowflakeInput, un tLogRow et un tSnowflakeClose dans le Job.
3. Reliez le composant tFixedFlowInput au tSnowflakeOutput à l’aide d’un lien Row > Main.
4. Répétez l’opération pour connecter le tSnowflakeClose au tLogRow.
5. Reliez le tSnowflakeConnection au tSnowflakeRow à l’aide d’un lien Trigger > On Subjob Ok.
6. Répétez l’opération pour relier le composant tSnowflakeRow au tFixedFlowInput, le tFixedFlowInput au tSnowflakeInput et le tSnowflakeInput au tSnowflakeClose.

Se connecter à Snowflake

Configurez le composant tSnowflakeConnection pour vous connecter à Snowflake.

Procédure

1. Double-cliquez sur le tSnowflakeConnection pour ouvrir sa vue Basic settings.
2. Dans le champ Account, saisissez le nom du compte assigné par Snowflake.
3. Dans les champs User Id et Password, saisissez les informations d’authentification.
   Notez que cet ID utilisateur est votre identifiant de connexion. Si vous ne connaissez pas votre identifiant, contactez l’administrateur de votre système Snowflake.
4. Dans le champ Warehouse, saisissez le nom de l’entrepôt de données à utiliser dans Snowflake.
5. Dans le champ Schema, saisissez le nom du schéma de la base de données à utiliser.
6. Dans le champ Database, saisissez le nom de la base de données à utiliser.

Créer une table Snowflake

Configurez le composant tSnowflakeRow pour créer une nouvelle table.

Procédure

1. Double-cliquez sur le tSnowflakeRow pour ouvrir sa vue Basic settings.
2. Dans la liste déroulante **Connection Component**, sélectionnez le composant **tSnowflakeConnection** afin de réutiliser la connexion créée par celui-ci. Dans cet exemple, sélectionnez **tSnowflakeConnection_1**.

3. Dans le champ **Query**, saisissez l'instruction SQL utilisée pour créer une nouvelle table dans laquelle écrire les données.

Dans cet exemple, l'instruction SQL suivante est utilisée pour créer ou remplacer une table **EMPLOYEE** contenant quatre colonnes, **ID** de type **INT**, **NAME** de type **VARCHAR**, **ONBOARD** de type **DATE** et **SALARY** de type **VARIANT**.

```sql
"CREATE OR REPLACE TABLE EMPLOYEE (" +
"ID INT NOT NULL primary key," +
"NAME VARCHAR (50), " +
"ONBOARD DATE," +
"SALARY VARIANT" +
") COMMENT = 'Created By Doc Team'"
```

**Écrire des données dans Snowflake**

Configurez le composant **tFixedFlowInput** et le **tSnowflakeOutput** pour écrire des données dans Snowflake.

**Procédure**

1. Double-cliquez sur le **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton [...] à côté du champ **Edit schema** et, dans la fenêtre de schéma qui s’ouvre, définissez le schéma en ajoutant quatre colonnes, **ID** de type **int**, **NAME** et **SALARY** de type **String** et **ONBOARD** de type **Date**.
3. Cliquez sur OK pour valider ces modifications et acceptez la propagation proposée par la boîte de dialogue.

4. Dans le champ **Number of rows**, saisissez le nombre d’enregistrements à générer, 5 dans cet exemple.

5. Dans la zone **Mode**, sélectionnez **Use Single Table** et spécifiez la valeur pour chaque colonne.
   - ID : numéro automatiquement incrémenté par la routine `Numeric.sequence("id",1,1)`.
   - NAME : prénom aléatoire généré par la routine `TalendDataGenerator.getFirstName()`.
   - ONBOARD : date aléatoire générée par la routine `TalendDate.getRandomDate("2007-09-01","2017-09-01").`
   - SALARY : données semi-structurées JSON au format `{"Salary": value}`, où la valeur du salaire est générée par la routine `Numeric.random(100000,200000)`.

6. Double-cliquez sur le composant **tSnowflakeOutput** pour ouvrir sa vue **Basic settings**.

7. Dans la liste déroulante **Connection Component**, sélectionnez le composant **tSnowflakeConnection** pour réutiliser la connexion créée par ce composant.

8. Cliquez sur le bouton [...] à côté du champ **Table** et, dans la boîte de dialogue qui s’ouvre, cochez la case **Use custom object** et saisissez dans le champ **Object Name** le nom de la table dans laquelle écrire les données. Dans cet exemple, saisissez `EMPLOYEE`, la table créée par le composant **tSnowflakeRow**. Cela fait, cliquez sur OK pour fermer la boîte de dialogue.


**Lire des données de Snowflake**

Configurez le composant **tSnowflakeInput** et le **tLogRow** afin de récupérer des données de Snowflake et les écrire dans la console.

**Procédure**

1. Double-cliquez sur le composant **tSnowflakeInput** pour ouvrir sa vue **Basic settings**.

2. Dans la liste déroulante **Connection Component**, sélectionnez le composant **tSalesforceConnection** afin de réutiliser la connexion créée par ce composant.

3. Cliquez sur le bouton [...] à côté du champ **Table** et, dans la boîte de dialogue qui s’ouvre, cochez la case **Use custom object**. Saisissez dans le champ **Object Name** le nom de la table Snowflake.
de laquelle récupérer les données. Dans cet exemple, saisissez EMPLOYEE, la table créée par le composant tSnowflakeRow. Cela fait, cliquez sur OK pour fermer la boîte de dialogue.

4. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la fenêtre de schéma qui s’ouvre, définissez le schéma en ajoutant quatre colonnes : ID de type int, NAME et SALARY de type String et \_<span class="hljs-variable language""><span class="hljs-title">ONBOARD</span></span> de type Date. Ce schéma est le même que le schéma du composant tSnowflakeOutput.

5. Cliquez sur OK afin de valider ces modifications et acceptez la propagation proposée par la boîte de dialogue qui s’ouvre.

6. Double-cliquez sur le composant tLogRow pour ouvrir sa vue Component.

7. Dans la zone Mode, sélectionnez Vertical (each row is a key/value list) pour une lisibilité optimale des résultats.

**Fermer la connexion à Snowflake**

Configurez le composant tSnowflakeClose pour fermer la connexion à Snowflake.

**Procédure**

1. Double-cliquez sur le composant tSnowflakeClose pour ouvrir l’onglet Component.

2. Sélectionnez le composant qui ouvre la connexion que vous avez besoin de fermer dans la liste déroulante Connection Component, tSnowflakeConnection_1 dans cet exemple.

**Exécuter le Job pour écrire et lire des données dans Snowflake**

Une fois le Job et ses composants configurés pour écrire et lire des données Snowflake, vous pouvez exécuter le Job et vérifier les résultats d’exécution.

**Procédure**

1. Appuyez sur les touches Ctrl + S afin de sauvegarder le Job.

2. Appuyez sur F6 pour exécuter le Job.

Dans ce schéma, les données sont écrites dans la table EMPLOYEE dans Snowflake, puis lues à partir de cette table et sont affichées en sortie dans la console de la vue Run du Studio.
```json
#1. tLogRow_1
ID   | 1
NAME | John
ONBOARD | 2016-07-22
SALARY | {
    "Salary": 132099
}

#2. tLogRow_1
ID   | 2
NAME | Rutherford
ONBOARD | 2009-04-27
SALARY | {
    "Salary": 138064
}

#3. tLogRow_1
ID   | 3
NAME | Millard
ONBOARD | 2012-06-04
SALARY | {
    "Salary": 165287
}

#4. tLogRow_1
ID   | 4
NAME |
ONBOARD | 
SALARY | 
```

**tSnowflakeOutput**

Ce composant utilise les données entrantes du composant précédent pour insérer (insert), mettre à jour (update), mettre à jour et insérer (upsert) ou supprimer (delete) des données dans une table Snowflake.

Il utilise la fonction de chargement de masse (bulk loader) fourni par Snowflake pour des opérations de base de données avec de hautes performances.

**Propriétés du tSnowflakeOutput Standard**

Ces propriétés sont utilisées pour configurer le tSnowflakeOutput s'exécutant dans le framework de Jobs Standard.

Le composant tSnowflakeOutput Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Paramètre</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. Cette propriété n’est pas disponible lorsqu’un autre composant de connexion est sélectionné dans la liste Connection Component.</td>
</tr>
<tr>
<td>Connection Component</td>
<td>Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant.</td>
</tr>
<tr>
<td>Account</td>
<td>Dans le champ Account, saisissez, entre guillemets doubles, le nom qui vous a été assigné par Snowflake.</td>
</tr>
</tbody>
</table>
| **User Id et Password** | Saisissez entre guillemets doubles, vos informations d’authentification à Snowflake.  
- Dans le champ **User ID**, saisissez, entre guillemets doubles, votre identifiant défini dans Snowflake via le paramètre `LOGIN_NAME` de Snowflake. Pour plus d’informations, contactez l’administrateur de votre système Snowflake.  
- Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warehouse</strong></td>
<td>Saisissez, entre guillemets doubles, le nom de l’entrepôt Snowflake à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Saisissez, entre guillemets doubles, le nom du schéma de la base de données à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez, entre guillemets doubles, le nom de la base de données Snowflake à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Cliquez sur le bouton [...] et, dans l’assistant qui s’affiche, sélectionnez la table Snowflake à utiliser.</td>
</tr>
</tbody>
</table>
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.  
**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
Si le type de données Snowflake à gérer est VARIANT, OBJECT ou ARRAY, lorsque vous définissez le schéma dans le composant, sélectionnez `String` pour les données correspondantes dans la colonne **Type** de l’éditeur de schéma.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. |
**Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Notez que, si la valeur d’entrée de tout champ primitif non nullable est nulle, la ligne de données comprenant ce champ sera rejetée.

### Output Action

Sélectionnez l’opération permettant d’insérer, supprimer, mettre à jour ou fusionner des données dans des tables Snowflake.

L’opération d’**Upsert** vous permet de fusionner des données dans une table Snowflake, à partir des données entrant dans le *tSnowflakeOutput*. Après avoir sélectionné **Upsert**, sélectionnez la colonne à utiliser comme clé de jointure pour cette opération.

### Advanced settings

<table>
<thead>
<tr>
<th>Login Timeout</th>
<th>Spécifiez le temps d’attente d’une réponse lors de la connexion à Snowflake, avant de retourner une erreur.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracing</td>
<td>Sélectionnez le niveau de log pour le pilote JDBC de Snowflake. Si cette option est activée, un log standard Java est généré.</td>
</tr>
<tr>
<td>Role</td>
<td>Saisissez, entre guillemets doubles, le rôle de contrôle des accès par défaut à utiliser pour initialiser la session Snowflake. Ce rôle doit déjà exister et doit avoir été assigné à l’ID de l’utilisateur que vous utilisez pour vous connecter Snowflake. Si vous laissez ce champ vide, le rôle PUBLIC est automatiquement assigné. Pour plus d’informations concernant le modèle de contrôle des accès de Snowflake, consultez la documentation Snowflake à l’adresse suivante <em>Understanding the Access Control Model</em> (en anglais).</td>
</tr>
<tr>
<td>Convert columns and table to uppercase</td>
<td>Cochez cette case pour passer les minuscules en majuscules, dans le nom de la table et des colonnes du schéma définies. Cette propriété n’est pas disponible lorsque la case Manual Query est cochée.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>
Variables globales

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_SUCCESS</td>
<td>Nombre de lignes traitées correctement. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_REJECT</td>
<td>Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Détails</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant est un composant de fin de flux de données dans votre Job. Il reçoit des données d’autres composants via un lien Row &gt; Main. Il peut également envoyer des messages d’erreur à d’autres composants via un lien Row &gt; Rejects. Les informations fournies à propos d’une erreur peuvent être :</td>
<td></td>
</tr>
<tr>
<td>• le nom de la colonne en erreur.</td>
<td></td>
</tr>
<tr>
<td>• le numéro de la ligne de données en erreur.</td>
<td></td>
</tr>
<tr>
<td>• la catégorie d’erreur, comme une erreur de parsing ou une erreur de conversion.</td>
<td></td>
</tr>
<tr>
<td>• l’offset du caractère dans la ligne de code en erreur.</td>
<td></td>
</tr>
<tr>
<td>• le message d’erreur associé.</td>
<td></td>
</tr>
<tr>
<td>• l’offset de l’octet en erreur.</td>
<td></td>
</tr>
<tr>
<td>• le numéro de la ligne de code en erreur.</td>
<td></td>
</tr>
<tr>
<td>• SQLSTATE de l’erreur.</td>
<td></td>
</tr>
<tr>
<td>• le code d’erreur Snowflake.</td>
<td></td>
</tr>
</tbody>
</table>

Scénario associé

Pour un scénario associé, consultez Écrire et lire des données dans une table Snowflake à la page 3763.
tSnowflakeRow

Ce composant exécute la commande SQL définie sur une base de données Snowflake spécifiée.

**Propriétés du tSnowflakeRow Standard**

Ces propriétés sont utilisées pour configurer le tSnowflakeRow s'exécutant dans le framework de Jobs Standard.

Le composant tSnowflakeRow Standard appartient à la famille Cloud.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td>Connection Component</td>
<td>Sélectionnez le composant établissant la connexion à la base de données à réutiliser par ce composant.</td>
</tr>
<tr>
<td>Account</td>
<td>Dans le champ Account, saisissez, entre guillemets doubles, le nom qui vous a été assigné par Snowflake.</td>
</tr>
<tr>
<td>User Id et Password</td>
<td>Saisissez entre guillemets doubles, vos informations d’authentification à Snowflake.</td>
</tr>
</tbody>
</table>
|            | - Dans le champ User ID, saisissez, entre guillemets doubles, votre identifiant défini dans Snowflake via le paramètre LOGIN_NAME de Snowflake. Pour plus
d’informations, contactez l’administrateur de votre système Snowflake.

- Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

<table>
<thead>
<tr>
<th>Warehouse</th>
<th>Saisissez, entre guillemets doubles, le nom de l’entrepôt Snowflake à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Saisissez, entre guillemets doubles, le nom du schéma de la base de données à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez, entre guillemets doubles, le nom de la base de données Snowflake à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.</td>
</tr>
<tr>
<td>Table</td>
<td>Cliquez sur le bouton [...] et, dans l’assistant qui s’affiche, sélectionnez la table Snowflake à utiliser.</td>
</tr>
</tbody>
</table>

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réserve line lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Si le type de données Snowflake à gérer est VARIANT, OBJECT ou ARRAY, lorsque vous définissez le schéma dans le composant, sélectionnez String pour les données correspondantes dans la colonne Type de l’éditeur de schéma.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la
<table>
<thead>
<tr>
<th><strong>Guess Query</strong></th>
<th>Cliquer sur le bouton afin de générer la requête correspondant à la table et au schéma, dans le champ <strong>Query</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query</strong></td>
<td>Spécifiez la commande SQL à exécuter. Pour plus d’informations concernant les commandes Snowflake SQL, consultez SQL Command Reference (en anglais).</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. Décrochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque des erreurs sont ignorées, vous pouvez collecter les lignes en erreur à l’aide d’un lien <strong>Row &gt; Reject</strong>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>Login Timeout</strong></th>
<th>Spécifiez le temps d’attente d’une réponse lors de la connexion à Snowflake, avant de retourner une erreur.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tracing</strong></td>
<td>Sélectionnez le niveau de log pour le pilote JDBC de Snowflake. Si cette option est activée, un log standard Java est généré.</td>
</tr>
<tr>
<td><strong>Role</strong></td>
<td>Saisissez, entre guillemets doubles, le rôle de contrôle des accès par défaut à utiliser pour initialiser la session Snowflake. Ce rôle doit déjà exister et doit avoir été assigné à l’ID de l’utilisateur que vous utilisez pour vous connecter Snowflake. Si vous laissez ce champ vide, le rôle PUBLIC est automatiquement assigné. Pour plus d’informations concernant le modèle de contrôle des accès de Snowflake, consultez la documentation Snowflake à l’adresse suivante Understanding the Access Control Model (en anglais).</td>
</tr>
<tr>
<td><strong>Propagate QUERYs recordset</strong></td>
<td>Cochez cette case pour propager le résultat de la requête SELECT dans le flux de sortie.</td>
</tr>
</tbody>
</table>
| **Use PreparedStatement** | Cochez cette case si vous souhaitez interroger la base de données à l’aide d’une instruction préparée. Dans la table **Set PreparedStatement Parameters** qui s’affiche, spécifiez la valeur pour chaque paramètre représenté par un point d’interrogation ?., dans l’instruction SQL définie dans le champ **Query**.  
  - **Parameter Index** : position du paramètre dans l’instruction SQL.  
  - **Parameter Type** : type de données du paramètre.  
  - **Parameter Value** : valeur du paramètre. |
Pour un scénario utilisant cette propriété, consultez Scénario 2 : Utiliser l’instance PreparedStatement pour faire une requête sur des données à la page 2702.

<table>
<thead>
<tr>
<th>Commit every</th>
<th>Spécifiez le nombre de lignes à traiter avant de commiter un lot de lignes dans la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Dynamic settings**

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend. |

**Variables globales**

<table>
<thead>
<tr>
<th>NB_LINE</th>
<th>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

| Règle d’utilisation | Ce composant offre la flexibilité des requêtes de bases de données et couvre toutes les requêtes SQL possibles. |
Scénario associé

Pour un scénario associé, consultez Écrire et lire des données dans une table Snowflake à la page 3763.
tSOAP

Ce composant appelle une méthode via un service Web afin de récupérer les valeurs des paramètres définies dans l’éditeur du composant.

Le tSOAP envoie le message SOAP défini avec les paramètres donnés au service Web invoqué et retourne la valeur comme définie, à partir des paramètres donnés.

**Propriétés du tSOAP Standard**

Ces propriétés sont utilisées pour configurer le tSOAP s'exécutant dans le framework de Jobs Standard.

Le composant tSOAP Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Use NTLM</td>
<td>Cochez cette case si vous utilisez un protocole d'authentification NTLM. <strong>Domain</strong> : Nom de domaine du client.</td>
</tr>
<tr>
<td>Need authentication</td>
<td>Cochez la case d'authentification et renseignez le nom de l’utilisateur dans le champ Username et son mot de passe dans le champ Password, si cela est nécessaire pour accéder au service. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Use http proxy</strong></td>
<td>Cochez cette case si vous vous connectez derrière un proxy et renseignez les informations correspondantes.</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| **Trust server with SSL** | Cochez cette case pour authentifier le serveur auprès du client via un protocole SSL et renseignez les champs correspondants.  
  **TrustStore file** : saisissez le chemin d’accès et le nom du fichier TrustStore contenant la liste des certificats approuvés par le client.  
  **TrustStore password** : saisissez le mot de passe utilisé pour vérifier l’intégrité des données TrustStore. |
| **ENDPOINT** | Saisissez l’URL du serveur Web invoqué. |
| **SOAP action** | Saisissez l’URL de l’en-tête HTTP SOAPAction à utiliser pour identifier le but de la requête HTTP SOAP. |
| **SOAP version** | Sélectionnez la version du système SOAP que vous utilisez.  
  **Avertissement** :  
  L’enveloppe SOAP requise varie selon les versions. |
| **Use a message from the input schema** | Cochez cette case pour lire un message SOAP du composant précédent et l’envoyer au service Web invoqué.  
  Lorsque cette case est cochée, le champ **SOAP message** devient une liste déroulante, vous permettant de sélectionner le type de colonne **Document** afin de lire un fichier d’entrée XML.  
  **Avertissement** :  
  Il est logique d’utiliser cette option lorsque le composant **tSOAP** est relié à un composant d’entrée dont le schéma contient une colonne de type **Document** afin de lire un message SOAP valide. |
| **Output in Document** | Cochez cette case afin d’écrire le message de réponse au format XML. |
| **SOAP message** | Saisissez le message SOAP à envoyer au service Web invoqué. Les variables globales et de contexte peuvent être utilisées lorsque vous écrivez un message SOAP.  
  Pour plus d’informations concernant les variables de contexte, consultez le Guide utilisateur du Studio Talend. |

**Advanced settings**

| **Use Kerberos** | Cochez cette case pour sélectionner un composant **tSetKerberosConfiguration** dans la liste Kerberos configuration.  
  **Remarque** : |
|------------------|--------------------------------------------------------------------------------------------------|
Le lien Trigger > OnSubjobOk du composant tSetKerberosConfiguration doit être utilisé lors d’une connexion au tSOAP.

### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.

### Global Variables

**Global Variables**

- **ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

- Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

- Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

- Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**

Ce composant peut être utilisé en tant que composant d’entrée ou intermédiaire.

**Connections**

- Liens de sortie (de composant à un autre) :
  - **Row** : Main, Iterate.
  - **Trigger** : Run if, On Component Ok, On Component Error.

- Liens d’entrée (d’un autre composant à celui-ci) :
  - **Row** : Main, Iterate
  - **Trigger** : Run if, On Component Ok, On Component Error.

Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.

### Scénario : Récupérer le nom d’un pays en utilisant un Service Web

Ce scénario décrit un Job à deux composants utilisant un Service Web pour récupérer le nom d’un pays à partir d’un code pays.
Procédure

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : tSOAP et tLogRow.

2. Cliquez-droit sur le composant tSOAP afin d’ouvrir le menu contextuel. Sélectionnez Row > Main et cliquez sur le tLogRow pour relier les composants à l’aide d’un lien Row > Main.

3. Double-cliquez sur le tSOAP afin d’ouvrir sa vue Basic settings et définir ses propriétés.

4. Dans le champ ENDPOINT, saisissez ou collez l’URL du Service Web à utiliser, entre guillemets :
   "http://www.webservicex.net/country.asmx".


Remarque :

Vous pouvez voir cette adresse en regardant le WSDL du Service Web que vous appelez. Pour le Service Web de cet exemple, dans votre navigateur, saisissez ?wsdl à la fin de l’URL du Service Web utilisé dans le champ ENDPOINT, ouvrez la page Web correspondante, puis regardez l’action SOAPAction définie sous le nœud Operation :

<wsdl:operation name="GetCountryByCountryCode">
6. Dans le champ **SOAP version**, sélectionnez la version du système SOAP utilisé. Dans ce scénario, la version est **SOAP 1.1**.

7. Dans le champ **SOAP message**, saisissez le message, au format XML, utilisé pour récupérer les informations de pays du Service Web invoqué. Dans cet exemple, IS est utilisé comme code pays, le message est donc :

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
" xmlns:web="http://www.webserviceX.NET">
<soapenv:Header/>
<soapenv:Body>
<web:GetCountryByCountryCode>
  <!--Optional:-->
</web:GetCountryByCountryCode>
</soapenv:Body>
</soapenv:Envelope>
```

8. Sauvegardez votre Job et appuyez sur **F6** pour l’exécuter.

**Résultats**

Le nom du pays est récupéré et affiché dans la console de la vue **Run** à partir du code pays IS.

```xml
[statistics] connecting to socket on port 3879
[statistics] connected
  &lt;Table&gt;
  &lt;countrycode&gt;IS&lt;/countrycode&gt;
  &lt;name&gt;Iceland&lt;/name&gt;
  &lt;/Table&gt;
[statistics] disconnected
```

**Scénario 2 : Utiliser un message SOAP depuis un fichier XML pour obtenir le nom d’un pays et le sauvegarder dans un fichier XML**

Ce scénario décrit un Job à trois composants utilisant un message SOAP d’un fichier d’entrée XML afin d’invoquer un service Web. Il permet également récupérer le nom d’un pays correspondant à un code pays donné, dans cet exemple, IR, puis d’écrire la réponse dans un fichier XML.

**Déposer et relier les composants**

**Procédure**

1. Déposez les composants suivants de la **Palette** dans l’espace de modélisation graphique : un **tFileInputXML**, un **tSOAP** et un **tFileOutputXML**.
2. Connectez les composants à l’aide de liens **Main > Row**.
Configurer le composant d’entrée

Procédure

1. Double-cliquez sur le composant **tFileInputXML** pour ouvrir sa vue **Basic settings** et configurer ses propriétés.

2. Cliquez sur le bouton [...] à côté du champ **Edit schema** afin d’ouvrir la boîte de dialogue **[Schema]**.

3. Cliquez sur le bouton [+] pour ajouter une colonne et nommez-la **getCountryName** dans ce scénario. Sélectionnez le type **Document** dans la liste **Type**, puis cliquez sur **OK** pour fermer la boîte de dialogue.

4. Dans le champ **File name/Stream**, saisissez le chemin d’accès au fichier d’entrée XML contenant le message SOAP à utiliser, ou parcourrez votre système jusqu’à ce fichier en cliquant sur le bouton [...]
Le fichier d’entrée contient le message SOAP suivant. Vous pouvez constater que le code pays donné est IR.

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:web="http://www.webserviceX.NET">
  <soapenv:Header/>
  <soapenv:Body>
    <web:GetCountryByCountryCode>
      <web:CountryCode>IR</web:CountryCode>
    </web:GetCountryByCountryCode>
  </soapenv:Body>
</soapenv:Envelope>
```

5. Dans le champ **Loop XPath query**, saisissez "/" afin de définir l’élément racine comme nœud de boucle de la structure du fichier d’entrée.

6. Dans la table **Mapping**, renseignez la colonne **XPath query** en saisissant "." pour extraire toutes les données du nœud contexte de la source. Cochez la case **Get Nodes** pour construire un flux de données de type **Document**.

**Configurer le service Web à l’aide du tSOAP**

**Procédure**

1. Double-cliquez sur le composant **tSOAP** pour ouvrir sa vue **Basic settings**.

2. Dans le champ **ENDPOINT**, saisissez ou collez l’URL du service Web à utiliser, entre guillemets :
   "http://www.webservicex.net/country.asmx".

3. Dans le champ **SOAP Action**, saisissez ou collez l’URL de l’en-tête HTTP SOAPAction indiquant que vous souhaitez récupérer les informations du pays : 

4. Cochez la case **Use a message from the input schema** et sélectionnez une colonne de type **Document** dans la liste **SOAP Message** afin de lire le message SOAP du fichier d’entrée et de l’envoyer au service Web. Dans cet exemple, le schéma d’entrée possède une seule colonne, **getCountryName**.

5. Cochez la case **Output in Document** pour écrire en sortie le message de réponse au format XML.
Configurer le composant de sortie

Procédure

1. Double-cliquez sur le composant tFileOutputXML pour ouvrir sa vue Basic settings.

2. Dans le champ File Name, saisissez le chemin d'accès au fichier XML de sortie.
3. Cochez la case Incoming record is a document afin de récupérer le flux de données entrant en tant que document XML. Notez qu'une liste Column list apparait, vous permettant de sélectionner une colonne de laquelle récupérer les données. Dans cet exemple, le schéma contient une seule colonne.

Exécuter le Job

Procédure

1. Appuyez sur les touches Ctrl+S pour sauvegarder votre Job.
2. Appuyez sur la touche F6 ou cliquez sur le bouton Run dans l'onglet Run pour exécuter le Job.
Les informations concernant le pays correspondant au code pays IR sont retournées et sauvegardées dans le fichier XML défini.

```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
 <soap:Body>
   <GetCountryByCountryCodeResult>
    </NewDataSet>
    </Table>
    </CountryCodeResult>
  </GetCountryByCountryCodeResponse>
 </soap:Body>
</soap:Envelope>
```
tSocketInput

Ce composant ouvre le port de socket et écoute les données entrantes.
Le tSocketInput est un composant d’écoute, permettant de passer des données via un port défini.

Propriétés du tSocketInput Standard

Ces propriétés sont utilisées pour configurer le tSocketInput s’exécutant dans le framework de Jobs Standard.
Le composant tSocketInput Standard appartient à la famille Internet.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host name</td>
<td>Nom ou adresse IP du serveur Hôte.</td>
</tr>
<tr>
<td>Port</td>
<td>Port d’écoute à ouvrir.</td>
</tr>
<tr>
<td>Timeout</td>
<td>Nombre de secondes avant fermeture du socket du port d’écoute.</td>
</tr>
<tr>
<td>Uncompress</td>
<td>Cochez cette case pour extraire les données si nécessaire.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décocochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Rejects.</td>
</tr>
<tr>
<td>Field separator</td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td>Row separator</td>
<td>Chaîne (ex : &quot;\n&quot;sous Unix) utilisée pour séparer les lignes.</td>
</tr>
<tr>
<td>Escape Char</td>
<td>Caractère de la ligne à éviter.</td>
</tr>
<tr>
<td>Text enclosure</td>
<td>Caractères utilisés pour entourer le texte.</td>
</tr>
</tbody>
</table>
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être Built-in ou distant dans le Repository.  
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• View schema : sélectionnez cette option afin de voir le schéma. |
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].


**Encoding**

 Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du studio Talend.
Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant ouvre un point d’accès au poste de travail ou au serveur. Ce composant est un composant de début de Job et s’arrête uniquement après expiration.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

**Scénario : Transférer des données vers un port d’écoute**

Le scénario suivant décrit un double Job dont le but est de transférer des données via un port d’écoute. Le composant Socket peut aussi permettre de contrôler la communication entre les serveurs ne pouvant pas communiquer directement entre eux.

Créez deux Jobs : le premier (*SocketInput*) ouvre un port d’écoute et attend que les données soient envoyées. Le deuxième (*SocketOutput*) transmet des données délimitées d’un fichier vers le numéro de port correspondant au port d’écoute.

**Déposer et relier les composants**

**Procédure**

1. Dans le premier Job, cliquez et déposez les composants suivants dans l’espace numérique de travail : *tSocketInput* et *tLogRow*. Reliez à l’aide d’une connexion **Row> Main**

2. Dans le deuxième Job, cliquez et déposez les composants suivants de la **Palette** : *tFileInputDelimited* et *tSocketOutput*. Reliez à l’aide d’une connexion **Row> Main**.
Configurer les Jobs

Procédure

1. Dans le second Job, sélectionnez tFileInputDelimited et dans l’onglet Basic settings de la vue Component, configurez les paramètres d’accès au fichier d’entrée.

   - Property Type : Built-In
   - File name/Stream : "D:/Input/us_state.txt"
   - Row Separator : "\n"
   - Field Separator : ";"
   - Header
   - Row and Field separators
   - Header

   ![Component](image)

2. Dans le champ File Name, renseignez le chemin d’accès au fichier. Renseignez les séparateurs de lignes et de champs respectivement dans les champs Row et Field separators, ainsi que l’en-tête dans le champ Header.

3. Décrivez le schéma des données à passer au composant tSocketOutput.

   ![tFileInputDelimited](image)

4. Sélectionnez le composant tSocketOutput et configurez les paramètres de l’onglet Basic settings de la vue Component.

   - Host : 127.0.0.1
   - Port : 3456
   - Compress
   - Retry Times : 10
   - Timeout : 1000
   - Row Separator : "\n"
   - Field Separator : ";"
   - Escape char
   - Text enclosure
   - Schema : Built-In
   - Encoding Type : IS0-8859-15

5. Dans le champ Host, renseignez l’adresse IP et dans le champ Port, renseignez le numéro du port vers lequel les données seront transférées.

6. Dans le champ Retry, définissez le nombre d’essais à effectuer et dans le champ Timeout renseignez la durée (en secondes) avant que le Job ne s’arrête automatiquement.

7. Maintenant, configurez les paramètres du premier Job (SocketInput) contenant le composant tSocketInput.
Dans le champ **Host**, renseignez l’adresse IP et dans le champ **Port**, renseignez le numéro du port d’écoute vers lequel les données sont transférées.

Dans le champ **Timeout**, définissez la durée (en secondes) avant que le Job ne s’arrête automatiquement.

Modifiez le schéma afin qu’il corresponde en tout point ou en partie au schéma du deuxième Job.

**Exécuter les Jobs**

**Procédure**

1. Appuyez sur **F6** pour exécuter le Job *SocketInput* en premier afin d’ouvrir le port d’écoute et de le préparer à recevoir les données.

2. Le résultat s’affiche dans la vue **Run**, ainsi que les informations d’ouverture du socket.

   *Starting job SocketInput at 17:53 04/02/2008*

   **socket connected**

   AL|Alabama|Montgomery|Eirmingham

   AK|Alaska|Juneau|Anchorage

   AZ|Arizona|Phoenix|Phoenix

   AR|Arkansas|Little Rock|Little Rock

   CA|California|Sacramento|Los Angeles

   CO|Colorado|Denver|Denver

   CT|Connecticut|Hartford|Bridgeport

   DE|Delaware|Dover|Wilmington

   FL|Florida|Tallahassee|Jacksonville

   GA|Georgia|Atlanta|Atlanta

   HI|Hawaii|Honolulu|Honolulu

   ID|Idaho|Boise|Boise

   IL|Illinois|Springfield|Chicago

   IN|Indiana|Indianapolis|Indianapolis

**Résultats**

Avant que le Job ne s’arrête, lancez l’autre Job (*SocketOutput*) afin de transmettre les données.
tSocketOutput

Ce composant envoie les données du flux d’entrée vers le port d’écoute du socket.
Le tSocketOutput écrit des données sur un port d’écoute.

**Propriétés du tSocketOutput Standard**

Ces propriétés sont utilisées pour configurer le tSocketOutput s’exécutant dans le framework de Jobs Standard.

Le composant tSocketOutput Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Nom ou adresse IP du serveur Hôte.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Port d’écoute à ouvrir</td>
</tr>
<tr>
<td><strong>Compress</strong></td>
<td>Cochez cette case pour archiver les données si nécessaire.</td>
</tr>
<tr>
<td><strong>Retry times</strong></td>
<td>Nombre d’essais avant que le Job ne soit mis en échec.</td>
</tr>
<tr>
<td><strong>Timeout</strong></td>
<td>Nombre de secondes avant fermeture du port d’écoute.</td>
</tr>
<tr>
<td><strong>Field separator</strong></td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td><strong>Row separator</strong></td>
<td>Chaîne (ex : &quot;\n&quot; sous Unix) séparant les lignes.</td>
</tr>
<tr>
<td><strong>Escape Char</strong></td>
<td>Caractère de la ligne à éviter.</td>
</tr>
<tr>
<td><strong>Text enclosure</strong></td>
<td>Caractères utilisés pour entourer le texte.</td>
</tr>
</tbody>
</table>

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma peut être **Built-in** ou distant dans le **Repository**.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job...
**Encoding**

Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données des bases de données.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Global Variables**

**Global Variables**

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

**Utilisation**

**Règle d’utilisation**

Ce composant ouvre un point d’accès au poste de travail ou au serveur. Ce composant est un composant de début de Job et s’arrête uniquement après expiration.

**Limitation**

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans

Scénario associé

Pour un exemple d’utilisation du tSocketOutput, consultez Scénario : Transférer des données vers un port d’écoute à la page 3788.
**tSortRow**

Ce composant établit des métriques et des tables de classification.

Ce composant trie les données d’entrée basées sur une ou plusieurs colonnes, selon un type de tri et un ordre.

**Propriétés du tSortRow Standard**

Ces propriétés sont utilisées pour configurer le tSortRow s’exécutant dans le framework de Jobs Standard.

Le composant tSortRow Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.
Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
Cliquez sur Sync columns pour récupérer le schéma du composant précédent dans le Job. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
<tr>
<td><strong>Criteria</strong></td>
<td>Cliquez sur [+] pour ajouter autant de lignes que nécessaire pour mettre en place le tri. La première</td>
</tr>
</tbody>
</table>

3794
colonne définie dans le schéma est sélectionnée par défaut.

<table>
<thead>
<tr>
<th>Schema column</th>
<th>Sélectionnez la colonne de votre schéma sur laquelle vous souhaitez baser votre tri. Notez que l’ordre est important car il détermine la priorité de tri.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort type</td>
<td>Numérique ou Alphabétique.</td>
</tr>
<tr>
<td>Order</td>
<td>Ordre ascendant ou descendant.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Sort on disk</th>
<th>Personnalisez la mémoire utilisée de manière temporaire pour stocker les données de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temp data directory path</td>
<td>Définissez le chemin d’accès au dossier dans lequel stocker les fichiers temporaires.</td>
</tr>
<tr>
<td>Create temp data directory if not exists</td>
<td>Cochez cette case afin de créer le dossier s’il n’existe pas.</td>
</tr>
<tr>
<td>Buffer size of external sort</td>
<td>Entrez la taille de la mémoire physique à allouer au processus de tri.</td>
</tr>
</tbody>
</table>

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant.                                                                                                                                  |

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant manipule le flux de données et par conséquent requiert un composant en entrée et un composant en sortie. Il s’agit donc d’un composant intermédiaire.</th>
</tr>
</thead>
</table>
Trier des données

Le scénario suivant décrit un Job constitué de trois composants. Un composant **tRowGenerator** est utilisé pour créer des entrées de façon aléatoire. Ces entrées seront ensuite envoyées au composant **tSortRow** afin d’être triées selon une valeur définie. Dans ce scénario, le flux d’entrée contient des noms de vendeurs ainsi que leur volume de vente respectif et leur nombre d’années d’ancienneté dans l’entreprise. Le résultat de l’opération de tri est affiché dans la console **Run**.

- Cliquez et déposez les trois composants requis pour ce scénario : **tRowGenerator**, **tSortRow** et **tLogRow**.
- Connectez-les à l’aide de liens de type **Row Main**.
- Dans le **RowGenerator editor**, définissez les valeurs qui seront créées de manière aléatoire et qui seront ensuite triées par le **tSortRow**. Pour plus d’informations concernant l’utilisation du **tRowGenerator**, consultez tRowGenerator à la page 3478.

- Dans ce scénario, chaque vendeur est classé en fonction de la valeur de ses ventes (**Sales**) et de son ancienneté dans l’entreprise (**YearsInComp**).
- Double-cliquez sur **tSortRow** pour afficher l’onglet **Basic settings**. Définissez la priorité de tri sur la valeur des ventes et, en second critère, sur l’ancienneté.

- Utilisez le bouton [+ ] pour ajouter le nombre de lignes de critères requis. Paramétrez le type de tri, dans cet exemple, les deux critères sont de type numérique. Enfin, étant donné que la sortie est une classification, définissez l’ordre de tri comme descendant.
Dans l'onglet **Advanced settings**, cochez l'option **Sort on disk** pour modifier les paramètres de la mémoire temporaire. Dans le champ **Temp data directory path**, renseignez le chemin d'accès au dossier dans lequel vous voulez stocker les données temporaires. Dans le champ **Buffer size of external sort**, définissez la taille maximale de la mémoire tampon à allouer au traitement des données.

**Avertissement :**

La valeur par défaut de la mémoire tampon est de 100000 mais plus vous traitez un nombre important de lignes et/ou colonnes, plus cette valeur devra être élevée pour éviter l'interruption automatique du Job signifiée par le message d'erreur “out of memory”.

Assurez-vous que ce flux est connecté au composant de sortie **tLogRow**, afin d’afficher le résultat dans la console **Run**.

Appuyez sur **F6** pour exécuter le Job ou passez à la vue **Run** et cliquez sur le bouton **Run**. Le classement est d’abord basé sur la valeur des ventes puis sur le nombre d’années d’ancienneté.
tSplitRow

Ce composant vous permet de séparer une ligne d'entrée en plusieurs lignes de sortie.

Propriétés du tSplitRow Standard

Ces propriétés sont utilisées pour configurer le tSplitRow s'exécutant dans le framework de Jobs Standard.
Le composant tSplitRow Standard appartient à la famille Processing.
Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (Built-in), soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
* View schema : sélectionnez cette option afin de voir le schéma.
* Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
* Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. Cliquez sur Sync columns pour récupérer le schéma du composant précédent dans le Job. |


| Columns mapping | Cliquez sur le bouton [+] pour ajouter autant de lignes que nécessaire pour effectuer le mapping des colonnes d’entrée vers les colonnes de sortie. |
Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
|                 | NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. |
|                 | Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |
|                 | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
|                 | Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

Utilisation

| Règle d’utilisation | Ce composant découpe une ligne d’entrée en de multiples lignes de sortie, en mappant les colonnes d’entrée vers les colonnes de sortie. |

Scénario 1 : Séparer une ligne en plusieurs lignes

Ce scénario décrit un Job a trois composants. Une ligne de données contenant des informations concernant deux entreprises sera séparée en deux lignes.

![Diagramme de scénario 1](image)

Procédure

<table>
<thead>
<tr>
<th>Procédure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : tFixedFlowInput, tSplitRow et tLogRow.</td>
</tr>
<tr>
<td>2. Reliez-les à l’aide de liens Row Main.</td>
</tr>
</tbody>
</table>

4. Sélectionnez Use Inline Content(delimited file) dans la zone Mode.

5. Dans la zone Content, saisissez les scripts suivants :

   Talend;LA;California;537;5thAvenue;IT;Lionbridge;Memphis;Tennessee;537;Lincoln Road;IT Service;

6. Cliquez sur Edit schema pour ouvrir une boîte de dialogue afin d’éditer le schéma des données d’entrée.


8. Cliquez sur OK pour fermer la boîte de dialogue.

10. Cliquez sur **Edit schema** pour configurer le schéma des données de sortie.


12. Cliquez sur **OK** pour fermer la boîte de dialogue. Une table vide contenant le nom des colonnes définies dans l’étape précédente apparaît dans la zone **Columns mapping** :

13. Cliquez sur le bouton **[+]** sous la table **Columns mapping** afin d’ajouter deux lignes de sortie.

14. Renseignez la table **Columns mapping** en saisissant les valeurs suivantes dans les colonnes :

   * **Industry**: row1.Industry, row1.Industry2;
Remarque :
La valeur dans la colonne Address, par exemple, row1.Street","+row1.City","+row1.State, affiche une adresse absolue en combinant les valeurs des colonnes Street, City et State. "row1" utilisé dans les valeurs de chaque colonne fait référence à la ligne d’entrée du tFixedFlowInput.

15. Double-cliquez sur le tLogRow afin d’ouvrir sa vue Basic settings.

16. Cliquez sur le bouton Sync columns pour récupérer le schéma défini dans le composant précédent.
17. Sélectionnez Table (print values in cells of a table) dans la zone Mode.

Résultats
[statistics] connecting to socket on port 3706
[statistics] connected

<table>
<thead>
<tr>
<th>Company</th>
<th>CountryCode</th>
<th>Address</th>
<th>Industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telend</td>
<td>537</td>
<td>5th Avenue LA, California</td>
<td>IT</td>
</tr>
<tr>
<td>Lionbridge</td>
<td>537</td>
<td>Lincoln Road, Memphis, Tennessee</td>
<td>IT Service</td>
</tr>
</tbody>
</table>

[statistics] disconnected
Job Split ended at 18.21 27/10/2011. [exit code=0]

Les données d’entrée en une ligne sont séparées en deux lignes différentes, chaque ligne contenant des informations d’une des deux entreprises.
Ce composant envoie des données relatives aux événements à Splunk via son collecteur d'événements HTTP.
Le tSplunkEventCollector collecte et envoie les données relatives aux événements à Splunk.

**Propriétés du tSplunkEventCollector Standard**

Ces propriétés sont utilisées pour configurer le tSplunkEventCollector s'exécutant dans le framework de Jobs Standard.
Le composant tSplunkEventCollector Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Schema et Edit schema</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>• Built-In :</strong> Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
<tr>
<td><strong>• Repository :</strong> Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
<tr>
<td>Notez que le schéma de ce composant a été configuré par défaut avec les champs suivants. Vous pouvez cliquer sur le bouton [...] à côté du champ Edit schema pour visualiser et modifier le schéma prédéfini.</td>
<td></td>
</tr>
<tr>
<td><strong>• time</strong> : date/heure de l'événement. Notez que les données d'entrée sont au format de date Java et qu'elles sont transformées au format temps epoch requis par Splunk avant l'envoi au collecteur d'événements HTTP de Splunk.</td>
<td></td>
</tr>
<tr>
<td><strong>• source</strong> : valeur source des données relatives aux événements. Il s'agit généralement du chemin d'accès au fichier ou répertoire, du port réseau ou du script duquel provient l'événement.</td>
<td></td>
</tr>
<tr>
<td><strong>• sourcetype</strong> : type de source des données relatives aux événements. Il indique quel est le type de données.</td>
<td></td>
</tr>
<tr>
<td><strong>• host</strong> : hôte des données relatives aux événements. Il s'agit généralement du nom de l'hôte, de l'adresse IP ou du nom du domaine entièrement renseigné de la machine réseau dans laquelle provient l'événement.</td>
<td></td>
</tr>
<tr>
<td><strong>• index</strong> : nom de l'index par lequel les données relatives aux événements ont été indexées. Il doit</td>
<td></td>
</tr>
</tbody>
</table>
 faire partie de la liste des index autorisés si le paramètre indexes est configuré pour le jeton.

Pour plus d’informations sur le format des données relatives aux événements envoyées au collecteur d’événements HTTP de Splunk, consultez About the JSON event protocol in HTTP Event Collector (en anglais).

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Splunk Server URL

Saisissez l’URL utilisée pour accéder au serveur Web de Splunk.

### Token

Spécifiez le jeton du collecteur d’événements utilisé pour l’authentification des données relatives aux événements. Pour plus d’informations, consultez HTTP Event Collector token management (en anglais).

### Advanced settings

**Extended output**

Cochez cette case pour envoyer les données relatives aux événements à Splunk en mode batch. Dans le champ affiché, saisissez le nombre d’événements à traiter dans chaque batch.

Par défaut, cette case est cochée et le nombre d’événements à traiter dans chaque batch est 100.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

**Global Variables**

- **NB_LINE** : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.
- **RESPONSE_CODE** : code de réponse de Splunk. Cette variable est une variable After et retourne une chaîne de caractères.
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

## Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de fin dans un Job ou un sous-job et nécessite un lien d’entrée. |

## Scénario associé

Aucun scénario n’est disponible pour ce composant.
tSPSSInput

Ce composant envoie des données SPSS pour les écrire, dans un autre fichier par exemple.
Le tSPSSInput lit les données de fichiers SPSS (.sav).

Propriétés du tSPSSInput Standard

Ces propriétés sont utilisées pour configurer le tSPSSInput s’exécutant dans le framework de Jobs Standard.
Le composant tSPSSInput Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Sync schema</th>
<th>Cliquez sur ce bouton pour synchroniser le schéma avec les colonnes du fichier SPSS d’entrée.</th>
</tr>
</thead>
</table>
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Pour ce composant, le schéma des métadonnées est directement récupéré à partir du fichier SPSS d’entrée, c’est pourquoi il est en lecture seule.
Vous pouvez cliquer sur Edit schema pour voir les métadonnées qui ont été récupérées. |
| File name | Indiquez le nom ou le chemin d’accès au fichier SPSS à lire. |
| Translate labels | Cochez cette case pour traduire les étiquettes (labels) des valeurs stockées.
Remarque :
Si vous cochez cette case, vous devrez à nouveau récupérer les métadonnées. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables

| Global Variables | NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est |

une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est utilisé comme composant de début. Il requiert un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Avertissement:</strong></td>
</tr>
<tr>
<td></td>
<td>Avant de pouvoir utiliser toutes les fonctionnalités des composants SPSS, assurez-vous de suivre les étapes suivantes : -Si vous avez déjà installé SPSS, ajoutez le chemin du répertoire SPSS, qui se présente comme suit : <em>SET PATH=%PATH%;&lt;DR&gt;:\program\SPSS</em> . -Si vous n’avez pas encore installé SPSS, copiez le lib SPSS IO &quot;spssio32.dll&quot; à partir du CD d’installation et copiez-le dans votre répertoire “system32”.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Limitation</th>
<th>Oracle fournit deux types de plateformes JVM (32 bits et 64 bits). Par défaut, la JVM utilisée dans un système d’exploitation 64 bits est la version 64 bits de cette JVM. Le fichier JSPSS.dll utilisé par ce composant est compilé dans la version 32 bits de la JVM. Il est donc nécessaire de configurer la JVM 32 bits pour exécuter un Job dans un <em>Studio Talend</em>, installé sous un système d’exploitation 64 bits. Pour ce faire, procédez comme suit :</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Ouvrez le Job dans l’espace de modélisation graphique.</td>
</tr>
<tr>
<td>2.</td>
<td>Cliquez sur l’onglet <em>Run</em> et cliquez sur l’onglet <em>Advanced settings</em>.</td>
</tr>
<tr>
<td>3.</td>
<td>Cochez la case <em>Use specific JVM arguments</em> et cliquez sur le bouton <em>New</em>....</td>
</tr>
<tr>
<td>4.</td>
<td>Dans le champ <em>Set the VM argument</em>, saisissez <em>-d32</em> et cliquez sur <em>OK</em> afin de valider.</td>
</tr>
</tbody>
</table>
Scénario : Afficher le contenu d'un fichier SPSS

Le scénario suivant crée un Job à deux composants qui permet de lire chaque ligne d’un fichier SPSS (.sav) et de les afficher en sortie dans la console de log de la vue Run.

Construire le Job

Procédure
1. A partir de la Palette, déposez les composants tSPSSInput et tLogRow dans l’espace graphique.

   ![Diagramme de construction du Job](image)

2. Cliquez-droit sur le composant tPSSInput et connectez-le au tLogRow à l’aide d’un lien de type Main Row.

Configurer le composant d’entrée

Procédure
1. Double-cliquez sur le composant tSPSSInput pour afficher la vue Basic settings et configurer ses paramètres.

   ![Vue des paramètres Basic settings](image)

2. Cliquez sur le bouton [...] à côté du champ Filename pour parcourir vos dossiers jusqu’au fichier SPSS que vous voulez lire.


   ![Message de récupération de schéma](image)

4. Cliquez sur le bouton Yes pour fermer le message.
5. Si nécessaire, cliquez sur le bouton [...] à côté du champ **Edit schema** pour visualiser la structure de données prédéfinie dans le fichier SPSS source.

![](image)

6. Cliquez sur le bouton **OK** pour fermer la boîte de dialogue.

**Exécuter du Job**

Sauvegardez votre Job, et appuyez sur **F6** afin de l’exécuter.

Le fichier SPSS est lu ligne par ligne, et les champs extraits sont affichés dans la console.

```
Starting job SPSS at 10:28 20/01/2010
[statistics] connecting to socket on port 3802
[statistics] connected
1.0|1.0|52997.0|198522.0
2.0|1.0|57113.0|203200.0
3.0|1.0|54123.0|200744.0
1.0|0.0|63936.0|197791.0
2.0|0.0|64835.0|195714.0
3.0|0.0|66804.0|208239.0
[statistics] disconnected
Job SPSS ended at 10:28 20/01/2010. [exit code=0]
```

**Convertir les valeurs stockées**

**Pourquoi et quand exécuter cette tâche**

Afin d’effectuer de convertir les valeurs stockées, procédez comme suit :

**Procédure**

1. Dans la vue **Basic settings**, cochez la case **Translate label** si vous voulez traduire les étiquettes (labels) des valeurs stockées.

2. Cliquez à nouveau sur le bouton **Sync Schema**, afin de récupérer le schéma après traduction.

   Un message apparaît, vous demandant si vous souhaitez récupérer le schéma du fichier SPSS défini.

3. Cliquez sur **Yes (Oui)** pour fermer le message et procéder à l’étape suivante.

   Un second message s’ouvre et vous demande si vous souhaitez propager les modifications.

4. Cliquez sur **Yes (Oui)** afin de fermer le message et passer à l’étape suivante.

5. Enregistrez le Job puis appuyez sur la touche **F6** pour l’exécuter.
Résultats

Le fichier SPSS est lu ligne par ligne et les champs extraits sont affichés sur la console de log de la vue Run, après conversion des valeurs stockées.

Starting job SPSS at 10:32 20/01/2010.

| [statistics] connecting to socket on port 3418     |
| [statistics] connected                            |
| Under 21 | Female | 57997.0 | 198522.0 |
| 21-25     | Female | 57113.0 | 203200.0 |
| 26-30     | Female | 54123.0 | 200744.0 |
| Under 21 | Male   | 63936.0 | 167791.0 |
| 21-25     | Male   | 64835.0 | 195714.0 |
| 26-30     | Male   | 66804.0 | 208239.0 |

[statistics] disconnected

Job SPSS ended at 10:32 20/01/2010. [exit code=0]
**tSPSSOutput**

Ce composant écrit ou ajoute des données à un fichier SPSS. Il crée les fichiers SPSS directement ou écrase ceux qui existent déjà.

Le tSPSSOutput écrit des données dans un fichier SPSS (.sav).

**Propriétés du tSPSSOutput Standard**

Ces propriétés sont utilisées pour configurer le tSPSSOutput s’exécutant dans le framework de Jobs Standard.

Le composant tSPSSOutput Standard appartient à la famille Business Intelligence.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Sync schema</th>
<th>Cliquez sur ce bouton pour synchroniser le schéma avec les colonnes du fichier SPSS d’entrée.</th>
</tr>
</thead>
</table>
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
  - **View schema** : sélectionnez cette option afin de voir le schéma.  
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. |
| **Built-in** | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| **Repository** | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |
| **Filename** | Nom ou chemin d’accès du fichier SPSS à à écrire. |
**Write Type**

Sélectionnez dans la liste le type d’action à effectuer :

- **Write**: écrit simplement les nouvelles données.
- **Append**: écrit les nouvelles données à la suite de celles existantes.

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  
Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

**Utilisation**

| Règle d’utilisation | Ce composant ne peut être utilisé comme composant de début. Il nécessite un flux d’entrée.  
**Avertissement** :  
Avant de pouvoir utiliser toutes les fonctionnalités des composants SPSS, assurez-vous de suivre les étapes suivantes :  
1. Si vous avez déjà installé SPSS, ajoutez le chemin du répertoire SPSS, qui se présente comme suit :
   ```bash
   SET PATH=\%PATH\%;<DR>:\program\SPSS.
   ```  
2. Si vous n’avez pas encore installé SPSS, copiez le lib SPSS IO "spssio32.dll" à partir du CD d’installation et copiez-le dans votre répertoire "system32". |

| Limitation | Oracle fournit deux types de plateformes JVM (32 bits et 64 bits). Par défaut, la JVM utilisée dans un système d’exploitation 64 bits est la version 64 bits de cette JVM. Le fichier JSPSS.dll utilisé par ce composant est compilé dans la version 32 bits de la JVM. Il est donc nécessaire |
de configurer la JVM 32 bits pour exécuter un Job dans un Studio Talend, installé sous un système d'exploitation 64 bits.

Pour ce faire, procédez comme suit :

1. Ouvrez le Job dans l'espace de modélisation graphique.
2. Cliquez sur l'onglet Run et cliquez sur l'onglet Advanced settings.
3. Cochez la case Use specific JVM arguments et cliquez sur le bouton New....
4. Dans le champ Set the VM argument, saisissez -d32 et cliquez sur OK afin de valider.

Scénario : Ecrire des données dans un fichier .sav

Ce scénario décrit un Job très simple qui écrit des données d'entrée dans un fichier .sav.

Construire le Job

Procédure

1. Glissez-déposez de la Palette dans l'espace de modélisation graphique un tRowGenerator et un tSPSSOutput.
2. Cliquez-droit sur le tRowGenerator, et connectez-le au tSPSSOutput, à l'aide d'un lien Row Main.

Configurer le composant d’entrée

Procédure

1. Dans l'espace de modélisation graphique, double-cliquez sur le tRowGenerator afin d'afficher sa vue Basic settings et ouvrir son éditeur. Vous pouvez définir votre schéma.
2. Cliquez sur le bouton [+ ] afin d’ajouter les colonnes que vous voulez écrire dans le fichier .sav.
3. Définissez le schéma, et configurez les paramètres des colonnes.

   **Avertissement :**
   Vérifiez que vous avez défini la longueur de vos colonnes. Dans le cas contraire, un message d’erreur s’affichera lors de la construction du Job.

4. Cliquez sur OK afin de valider votre schéma et fermer l’éditeur.

**Configurer le composant de sortie**

**Procédure**

1. Double-cliquez sur le tSPSSOutput afin d’afficher sa vue Component et définir ses propriétés.

2. Cliquez sur le bouton [...] à côté du champ Filename, et parcourrez votre répertoire jusqu’au fichier SPSS .sav dans lequel vous souhaitez écrire les données.

3. Cliquez sur le bouton [...] à côté du bouton Sync columns, afin de synchroniser les colonnes avec celles du composant précédent. Dans cet exemple, le schéma devant être inséré dans le fichier .sav est composé de deux colonnes : id et country.

4. Si nécessaire, cliquez sur le bouton Edit schema pour voir/éditer le schéma défini.

5. Dans la liste Write Type, sélectionnez Write (écrire) ou Append (écrire à la suite), pour simplement écrire les données d’entrée dans le fichier .sav, ou bien les écrire à la fin de ce même fichier.
**Exécuter le Job**

Sauvegardez votre Job et appuyez sur **F6** pour l’exécuter.

Les données générées par le **tRowGenerator** sont écrites dans le fichier .sav défini.
tSPSSProperties

Ce composant obtient des informations sur les propriétés principales d'un fichier SPSS défini.
Le tSPSSProperties décrit les propriétés d'un fichier SPSS (.sav) défini.

Propriétés du tSPSSProperties Standard

Ces propriétés sont utilisées pour configurer le tSPSSProperties s'exécutant dans le framework de Jobs Standard.
Le composant tSPSSProperties Standard appartient à la famille Business Intelligence.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Pour ce composant, le schéma des métadonnées est prédéfini, c’est pourquoi il est en lecture seule. Il correspond à la convention interne SPSS. Vous pouvez cliquer sur Edit schema pour voir les métadonnées prédéfinies.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Repository</td>
<td>Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>Filename</td>
<td>Nom ou chemin d’accès au fichier SPSS à traiter.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables

| Global Variables | IS_VALID_FILE : résultat de la validation du fichier. Cette variable est une variable After et retourne un booléen. ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |

3816
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
</tr>
</thead>
</table>
| Ce composant est utilisé comme composant de début. Il requiert un composant de sortie.  
| **Avertissement :**  
| Pour pouvoir utiliser toutes les fonctionnalités des composants SPSS, assurez-vous de suivre les étapes suivantes : - Si vous avez déjà installé SPSS, ajoutez le chemin du répertoire SPSS, qui se présente comme suit : `SET PATH= %PATH%;<DR>::program\SPSS`. - Si vous n’avez pas encore installé SPSS, copiez le lib SPSS IO "spssio32.dll“ à partir du CD d’installation et copiez-le dans votre répertoire "system32". |

### Limitation

Oracle fournit deux types de plateformes JVM (32 bits et 64 bits). Par défaut, la JVM utilisée dans un système d'exploitation 64 bits est la version 64 bits de cette JVM. Le fichier JSPSS.dll utilisé par ce composant est compilé dans la version 32 bits de la JVM. Il est donc nécessaire de configurer la JVM 32 bits pour exécuter un Job dans un **Studio Talend**, installé sous un système d’exploitation 64 bits.

Pour ce faire, procédez comme suit :

1. Ouvrez le Job dans l’espace de modélisation graphique.
2. Cliquez sur l’onglet **Run** et cliquez sur l’onglet **Advanced settings**.
3. Cochez la case **Use specific JVM arguments** et cliquez sur le bouton **New**.
4. Dans le champ **Set the VM argument**, saisissez `–d32` et cliquez sur **OK** afin de valider.

### Scénario associé

Pour des scénarios associés, consultez :

- **Lire les données maître depuis un hub MDM** à la page 2276
- **Scénario : Ecrire des données dans un fichier .sav** à la page 3813
tSPSSStructure

Ce composant récupère des informations concernant les variables au sein des fichiers .sav.

Le tSPSSStructure traite les variables contenues dans les fichiers .sav. Vous pouvez utiliser ce composant avec un tFileList pour rassembler les informations concernant les fichiers *.sav existants, afin d’analyser de manière plus approfondie les résultats ou de les vérifier.

Propriétés du tSPSSStructure Standard

Ces propriétés sont utilisées pour configurer le tSPSSStructure s’exécutant dans le framework de Jobs Standard.

Le composant tSPSSStructure Standard appartient à la famille Business Intelligence.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
Pour ce composant, le schéma des métadonnées est prédéfini, c’est pourquoi il est en lecture seule. Il correspond à la convention interne SPSS. Vous pouvez cliquer sur Edit schema pour voir les métadonnées prédéfinies.

| Built-in | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

| Filename | Nom ou chemin d’accès au fichier SPSS à traiter.

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.
Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant est utilisé comme composant de début. Il requiert un composant de sortie.

⚠️ Avertissement :
Avant de pouvoir utiliser toutes les fonctionnalités des composants SPSS, assurez-vous de suivre les étapes suivantes : - Si vous avez déjà installé SPSS, ajoutez le chemin du répertoire SPSS, qui se présente comme suit : `SET PATH=%PATH%;<DR>\program\SPSS`. - Si vous n’avez pas encore installé SPSS, copiez le lib SPSS IO "spssio32.dll" à partir du CD d’installation et copiez-le dans votre répertoire "system32".

Limitation

Oracle fournit deux types de plateformes JVM (32 bits et 64 bits). Par défaut, la JVM utilisée dans un système d'exploitation 64 bits est la version 64 bits de cette JVM. Le fichier JSPSS.dll utilisé par ce composant est compilé dans la version 32 bits de la JVM. Il est donc nécessaire de configurer la JVM 32 bits pour exécuter un Job dans un Studio Talend, installé sous un système d'exploitation 64 bits.

Pour ce faire, procédez comme suit :
1. Ouvrez le Job dans l'espace de modélisation graphique.
3. Cochez la case Use specific JVM arguments et cliquez sur le bouton New....
4. Dans le champ Set the VM argument, saisissez `-d32` et cliquez sur OK afin de valider.

Scénario associé

Pour des scénarios associés, consultez :

- Lire les données maître depuis un hub MDM à la page 2276.
Scénario : Ecrire des données dans un fichier .sav à la page 3813.
tSQLDWHBulkExec

Ce composant charge des données dans une table Azure SQL Data Warehouse depuis Azure Blob Storage ou Azure Data Lake Store.

Pour plus d’informations concernant le chargement de données dans Azure SQL Data Warehouse, consultez Conception de processus ELT pour Azure SQL Data Warehouse.

Propriétés du tSQLDWHBulkExec Standard

Ces propriétés sont utilisées pour configurer le tSQLDWHBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tSQLDWHBulkExec Standard appartient à deux familles : Cloud and Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cochez cette case et sélectionnez le composant de connexion adapté à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td></td>
<td>Notez que lorsqu’un Job contient un Job parent et un Job enfant, si vous devez partager une connexion existante entre ces deux niveaux, par exemple pour partager la connexion créée par le Job père au Job fils, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. au niveau du Job père, enregistrer la connexion à la base de données à partager dans la vue Basic settings du composant de connexion créant cette connexion à la base de données.</td>
</tr>
<tr>
<td></td>
<td>2. au niveau du Job fils, utiliser un composant de connexion dédié afin de lire cette connexion enregistrée.</td>
</tr>
<tr>
<td></td>
<td>Pour un exemple concernant le partage de connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>JDBC Provider</strong></td>
<td>Sélectionnez le fournisseur du pilote JDBC à utiliser.</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Spécifiez l’adresse IP ou le nom d’hôte de l’entrepôt Azure SQL Data Warehouse à utiliser.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Spécifiez le numéro du port d’écoute de l’entrepôt Azure SQL Data Warehouse à utiliser.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Saisissez le nom du schéma Azure SQL Data Warehouse.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Spécifiez le nom de l’entrepôt Azure SQL Data Warehouse à utiliser.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez les informations d’authentification utilisateur pour accéder à Azure SQL Data Warehouse. Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Additional JDBC Parameters</strong></td>
<td>Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30; pour une connexion à la base de données Azure SQL.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Spécifiez le nom de la table de SQL Data Warehouse dans laquelle charger les données.</td>
</tr>
</tbody>
</table>
| **Action on table** | Sélectionnez une opération à effectuer sur la table définie, vous pouvez effectuer l’une des opérations suivantes :
  - None : aucune opération n’est effectuée.
  - Drop and create table : la table est supprimée et créée à nouveau.
  - Create table : la table n’existe pas et est créée.
  - Create table if not exists : la table est créée si elle n’existe pas.
  - Drop table if exists and create : la table est supprimée si elle existe déjà et créée à nouveau.
  - Clear table : le contenu de la table est supprimé. Vous pouvez annuler cette opération.
  - Truncate table : le contenu de la table est supprimé. Vous ne pouvez pas annuler cette opération. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. |
- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.


Cliquez sur *Edit schema* pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre *[Repository Content]*.

### Azure Storage

Sélectionnez le type de stockage Azure dans lequel charger les données, *Blob Storage* ou *Data Lake Store*.

### Account Name

Saisissez le nom de votre compte pour accéder à Azure Blob Storage ou Azure Data Lake Store.

### Access key

Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte.

Cette propriété est disponible uniquement lorsque l’option *Blob Storage* est sélectionnée dans la liste déroulante *Azure Storage*.

### Container

Saisissez le nom du conteneur d’objets blob.

Cette propriété est disponible uniquement lorsque l’option *Blob Storage* est sélectionnée dans la liste déroulante *Azure Storage*.

### Authentication key

Saisissez la clé d’authentification requise pour accéder à votre Azure Data Lake Store.

Cette propriété est disponible uniquement lorsque l’option *Data Lake Store* est sélectionnée dans la liste déroulante *Azure Storage*.

### Client Id

Saisissez l’ID de votre application (également appelée ID du client).

Cette propriété est disponible uniquement lorsque l’option *Data Lake Store* est sélectionnée dans la liste déroulante *Azure Storage*. 
### OAuth 2.0 token endpoint
Copiez-collez l'endpoint du jeton OAuth 2.0 que vous pouvez obtenir dans la page App registrations. Cette propriété est disponible uniquement lorsque l'option Data Lake Store est sélectionnée dans la liste déroulante Azure Storage.

### Azure Storage Location
Spécifiez l'emplacement où est créé votre compte Azure Blob Storage ou Azure Data Lake Store.

### Advanced settings

**File format**
Sélectionnez le format de fichier définissant les données externes stockées dans votre Azure Blob Storage ou Azure Data Lake Store, Delimited Text, Hive RCFile, Hive ORC, ou Parquet. Pour plus d'informations concernant les formats de fichiers, consultez CREATE EXTERNAL FILE FORMAT.

**Field separator**
Spécifiez le(s) caractère(s) indiquant la fin de chaque champ, dans le fichier délimité. Cette propriété est disponible uniquement lorsque l'option Delimited Text est sélectionnée dans la liste déroulante File format.

**Enclosed by**
Cochez cette case et, dans le champ proche, spécifiez le caractère entourant la chaîne de caractères dans le fichier délimité. Cette propriété est disponible uniquement lorsque l'option Delimited Text est sélectionnée dans la liste déroulante File format.

**Date format**
Cochez cette case et, dans le champ proche, spécifiez le format personnalisé pour toutes les données d'heure et de date dans le fichier délimité. Pour plus d’informations concernant le format de date, consultez CREATE EXTERNAL FILE FORMAT. Cette propriété est disponible uniquement lorsque l'option Delimited Text est sélectionnée dans la liste déroulante File format.

**Use type default**
Cochez cette case pour stocker chaque valeur manquante à l’aide de la valeur par défaut du type de données de la colonne correspondante. Décocochez cette case pour stocker chaque valeur manquante dans le fichier délimité en tant que NULL. Cette propriété est disponible uniquement lorsque l'option Delimited Text est sélectionnée dans la liste déroulante File format.

**Serde Method**
Sélectionnez une méthode Hive de sérialisation et désérialisation.
Cette propriété est disponible lorsque l'option Hive RCFile est sélectionnée dans la liste déroulante File format.

**Compressed by**
Cochez cette case si les données externes sont compressées. Dans la liste déroulante qui s’affiche, sélectionnez la méthode de compression.

**Data import reject options**
Cochez cette case pour spécifier les options de rejet suivantes.
- **Reject type** : spécifiez comment traiter les lignes rejetées.
- **Value** : si le nombre de lignes rejetées dépasse la valeur spécifiée dans le champ Reject value, le chargement échoue.
- **Percentage** : si le pourcentage de lignes rejetées dépasse la valeur spécifiée dans le champ Reject value, le chargement échoue.
- **Reject value** : valeur de rejet selon le type de rejet. Pour un pourcentage, la valeur est celle du pourcentage, sans le symbole %.
- **Reject sample value** : nombre de lignes à tenter de récupérer avant le calcul du pourcentage des lignes rejetées.

Pour plus d’informations concernant les options de rejet, consultez CREATE EXTERNAL TABLE.

**Distribution Option**
Sélectionnez le modèle de partitionnement (sharding) utilisé pour distribuer les données dans la table, Round Robin, Hash, ou Replicate. Pour plus d’informations concernant les modèles de partitionnement supportés par Azure SQL Data Warehouse, consultez Azure SQL Data Warehouse - Architecture MPP (Massively Parallel Processing).

Cette propriété est disponible lorsqu’une option relative à la création de table est sélectionnée dans la liste déroulante Action on table.

**Distribution Column Name**
Nom de la colonne de distribution pour une table e distribution par hachage.

Cette propriété est disponible uniquement lorsque l’option Hash est sélectionnée dans la liste déroulante Distribution Option.

**Table Option**
Sélectionnez le type d’index de la table, Clustered Columnstore Index, Heap, ou Clustered Index. Pour plus d’informations, consultez Indexation de tables dans SQL Data Warehouse.

Cette propriété est disponible lorsqu’une option relative à la création de table est sélectionnée dans la liste déroulante Action on table.

**Index column(s)**
Spécifiez le nom d’une ou plusieurs colonne(s) clé(s) dans l’index. Si plusieurs colonnes sont spécifiées, séparez-les par une virgule.
Cette propriété est disponible uniquement lorsque l’option Clustered Index est sélectionnée dans la liste déroulante Table Option.

Partition

Cochez cette case pour spécifier les options de partitionnement suivantes :

- **Partition column name** : spécifiez le nom de la colonne utilisée pour partitionner la table.
- **Range** : spécifiez la manière d’inclure la valeur limite dans la plage de limite.
  - **Left** : la valeur limite est incluse dans la plage gauche de la limite.
  - **Right** : la valeur limite est incluse dans la plage droite de la limite.
- **Partition For Values** : spécifiez les valeurs (séparées par une virgule) utilisées pour la partition.

Pour plus d’informations concernant les partitions de tables, consultez Partitionnement de tables dans SQL Data Warehouse.

Cette propriété est disponible lorsqu’une option relative à la création de table est sélectionnée dans la liste déroulante Action on table.

TStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

Variables globales

<table>
<thead>
<tr>
<th>ERROR_MESSAGE</th>
<th>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE_INSERTED</td>
<td>Nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
</tbody>
</table>

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en tant que composant stand-alone dans un Job ou un sous-job.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Certaines fonctionnalités supportées par d’autres bases de données ne sont pas supportées par Azure SQL Data Warehouse. Pour plus d’informations, consultez Fonctionnalités de table non prises en charge.</td>
</tr>
</tbody>
</table>

Scénario associé

Aucun scénario n’est disponible pour ce composant.
tSQLDWHClose

Ce composant ferme une connexion active à une base de données Azure SQL Data Warehouse.

Propriétés du tSQLDWHClose Standard

Ces propriétés sont utilisées pour configurer le tSQLDWHClose s'exécutant dans le framework de Jobs Standard.
Le composant tSQLDWHClose Standard appartient à deux familles : Cloud et Databases.
Le composant de ce framework est toujours disponible.

Basic settings

| Component List | Sélectionnez dans la liste le composant tSQLDWHConnection ouvrant la connexion à fermer. |

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

Global Variables

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Azure SQL Data Warehouse, particulièrement avec le tSQLDWHConnection et le tSQLDWHCommit. |

| Dynamic settings | Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. |

| Limitation | Certaines fonctionnalités supportées par d’autres bases de données ne sont pas supportées par Azure SQL Data Warehouse. Pour plus d’informations, consultez Fonctionnalités de table non prises en charge. |

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
**tSQLDWHCommit**

Ce composant commute en une fois une transaction globale au lieu de commiter ligne par ligne ou lot par lot, vous permettant ainsi d’améliorer les performances.

Notez que si vous devez commiter chaque instruction en tant que transaction individuelle, vous devez utiliser la fonctionnalité Auto Commit disponible dans le composant de connexion.

**Propriétés du tSQLDWHCommit Standard**

Ces propriétés sont utilisées pour configurer le composant tSQLDWHCommit s’exécutant dans le framework de Jobs Standard.

Le composant tSQLDWHCommit Standard appartient à deux familles : Cloud et Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Component List</th>
<th>Sélectionnez dans la liste le composant tSQLDWHConnection si plusieurs connexions sont définies dans le Job.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close Connection</td>
<td>Cette case est cochée par défaut. Elle vous permet de fermer la connexion à une base de données, une fois le commit effectué. Décrochez cette case pour continuer à utiliser la connexion sélectionnée, une fois que le composant a terminé sa tâche. Si vous utilisez un lien Row &gt; Main pour relier le tSQLDWHCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection ou votre connexion sera fermée avant la fin de votre premier commit de ligne.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Global Variables**

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Azure SQL Data Warehouse, notamment le tSQLDWHConnection et le tSQLDWHRollback. |
**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez le **Guide utilisateur du Studio Talend**.

**Limitation**

Certaines fonctionnalités supportées par d’autres bases de données ne sont pas supportées par Azure SQL Data Warehouse. Pour plus d’informations, consultez **Fonctionnalités de table non prises en charge**.

---

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tSQLDWHConnection

Ce composant ouvre une connexion à une base de données Azure SQL Data Warehouse.

Propriétés du tSQLDWHConnection Standard

Ces propriétés sont utilisées pour configurer le tSQLDWHConnection s’exécutant dans le framework de Jobs Standard.

Le composant tSQLDWHConnection Standard appartient à deux familles : Cloud et Databases.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JDBC Provider</th>
<th>Sélectionnez le fournisseur du pilote JDBC à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Spécifiez l’adresse IP ou le nom d’hôte de l’entrepôt Azure SQL Data Warehouse à utiliser.</td>
</tr>
<tr>
<td>Port</td>
<td>Spécifiez le numéro du port d’écoute de l’entrepôt Azure SQL Data Warehouse à utiliser.</td>
</tr>
<tr>
<td>Schema</td>
<td>Saisissez le nom du schéma Azure SQL Data Warehouse.</td>
</tr>
<tr>
<td>Database</td>
<td>Spécifiez le nom de l’entrepôt Azure SQL Data Warehouse à utiliser.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d’authentification utilisateur pour accéder à Azure SQL Data Warehouse.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Additional JDBC Parameters</td>
<td>Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple,</td>
</tr>
</tbody>
</table>
Use or register a shared DB Connection

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ Shared DB Connection Name qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options Use dynamic job et Use an independent process to run subjob du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.

Cette case est indisponible lorsque la case Specify a data source alias est cochée.

Specify a data source alias

Cochez cette case et, dans le champ Data source alias qui s’affiche, spécifiez l’alias d’une source de données créée du côté Talend Runtime, pour utiliser le pool de connexions partagées défini dans la configuration de la source de données. Cette option fonctionne uniquement lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour un scénario associé, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647.

Cette case est indisponible lorsque la case Use or register a shared DB Connection est cochée.

Advanced settings

Auto Commit

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commit pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour
gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

<table>
<thead>
<tr>
<th>Share identity insert in multi table</th>
<th>Cochez cette case pour partager IDENTITY_INSERT avec plusieurs tables au sein d’une connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>ERROR_MESSAGE</th>
<th>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec d’autres composants Azure SQL Data Warehouse. Il ouvre une connexion réutilisable par ces derniers.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Certaines fonctionnalités supportées par d’autres bases de données ne sont pas supportées par Azure SQL Data Warehouse. Pour plus d’informations, consultez Fonctionnalités de table non prises en charge.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tSQLDWHInput

Ce composant lit des données et extrait des champs en se basant sur une requête, à partir d’une base de données Azure SQL Data Warehouse.

Propriétés du tSQLDWHInput Standard

Ces propriétés sont utilisées pour configurer le tSQLDWHInput s’exécutant dans le framework de Jobs Standard.

Le tSQLDWHInput Standard appartient à deux familles : Cloud et Databases.

Le composant de ce framework est toujours disponible.

Basic settings

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, si vous devez partager une connexion existante entre ces deux niveaux, par exemple pour partager la connexion créée par le Job père au Job fils, vous devez :

1. au niveau du Job père, enregistrer la connexion à la base de données à partager dans la vue Basic settings du composant de connexion créant cette connexion à la base de données.

2. au niveau du Job fils, utiliser un composant de connexion dédié afin de lire cette connexion enregistrée.

Pour un exemple concernant le partage de connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend. |

| Property Type | Sélectionnez la manière de configurer les informations de connexion.

- **Built-In** : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.

- **Repository** : les informations de connexion stockées centralement dans le Repository > Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. |

<p>| JDBC Provider | Sélectionnez le fournisseur du pilote JDBC à utiliser. |</p>
<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Spécifiez l'adresse IP ou le nom d'hôte de l'entrepôt Azure SQL Data Warehouse à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Spécifiez le numéro du port d'écoute de l'entrepôt Azure SQL Data Warehouse à utiliser.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Saisissez le nom du schéma Azure SQL Data Warehouse.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Spécifiez le nom de l'entrepôt Azure SQL Data Warehouse à utiliser.</td>
</tr>
</tbody>
</table>
| **Username** et **Password** | Saisissez les informations d'authentification utilisateur pour accéder à Azure SQL Data Warehouse.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.  
- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**. |
| **Table name** | Spécifiez le nom de la table à utiliser. |
| **Query Type** | Sélectionnez la manière de configurer la requête.  
- **Built-in** : saisissez manuellement l'instruction de requête ou construisez-la graphiquement à l'aide du SQLBuilder. |
**Repository** : Sélectionnez la requête correspondant, stockée dans le Repository. Le champ Query est renseigné.

<table>
<thead>
<tr>
<th>Guess Query</th>
<th>Cliquez sur ce bouton pour générer la requête correspondant au schéma de votre table, dans le champ Query.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guess schema</td>
<td>Cliquez sur ce bouton pour générer les colonnes du schéma en se basant sur la requête spécifiée dans le champ Query.</td>
</tr>
<tr>
<td>Query</td>
<td>Spécifiez la requête en faisant particulièrement attention à mettre les champs dans un ordre correct, afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td>Specify a data source alias</td>
<td>Cochez cette case et, dans le champ Data source alias qui s’affiche, spécifiez l’alias d’une source de données créée du côté Talend Runtime, pour utiliser le pool de connexions partagées défini dans la configuration de la source de données. Cette option fonctionne uniquement lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour un scénario associé, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647. Cette case est indisponible lorsque la case Use an existing Connection est cochée.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Additional JDBC Parameters</th>
<th>Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30; pour une connexion à la base de données Azure SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim all the String/Char columns</td>
<td>Cochez cette case pour supprimer les espaces blancs en début et fin de champs de toutes les colonnes String/Char.</td>
</tr>
<tr>
<td>Trim column</td>
<td>Cochez les cases de la colonne Trim afin de supprimer les espaces blancs en début et fin de champ des colonnes correspondantes. Cette propriété est indisponible lorsque la case Trim all the String/Char columns est cochée.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>
Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>QUERY</td>
<td>Instruction de requête en cours de traitement. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie.</td>
<td></td>
</tr>
</tbody>
</table>

Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Limitation

Certaines fonctionnalités supportées par d’autres bases de données ne sont pas supportées par Azure SQL Data Warehouse. Pour plus d’informations, consultez Fonctionnalités de table non prises en charge.

Scénario associé

Aucun scénario n’est disponible pour ce composant.
**tSQLDWHOutput**

Ce composant écrit, met à jour, modifie ou supprime des entrées dans une base de données Azure SQL Data Warehouse.

**Propriétés du tSQLDWHOutput Standard**

Ces propriétés sont utilisées pour configurer le tSQLDWHOutput s’exécutant dans le framework de Jobs Standard.

Le composant tSQLDWHOutput Standard appartient à deux familles : Cloud et Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie. 

Notez que lorsqu’un Job contient un Job parent et un Job enfant, si vous devez partager une connexion existante entre ces deux niveaux, par exemple pour partager la connexion créée par le Job père au Job fils, vous devez :

1. au niveau du Job père, enregistrer la connexion à la base de données à partager dans la vue Basic settings du composant de connexion créant cette connexion à la base de données.

2. au niveau du Job fils, utiliser un composant de connexion dédié afin de lire cette connexion enregistrée.

Pour un exemple concernant le partage de connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend. |
| --- | --- |
| Property Type | Sélectionnez la manière de configurer les informations de connexion. 

- **Built-In** : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.

- **Repository** : les informations de connexion stockées centralement dans le Repository > Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. |
| JDBC Provider | Sélectionnez le fournisseur du pilote JDBC à utiliser. |
| **Host** | Spécifiez l'adresse IP ou le nom d'hôte de l'entrepôt Azure SQL Data Warehouse à utiliser. |
| **Port** | Spécifiez le numéro du port d'écoute de l'entrepôt Azure SQL Data Warehouse à utiliser. |
| **Schema** | Saisissez le nom du schéma Azure SQL Data Warehouse. |
| **Database** | Spécifiez le nom de l'entrepôt Azure SQL Data Warehouse à utiliser. |
| **Username et Password** | Saisissez les informations d'authentification utilisateur pour accéder à Azure SQL Data Warehouse. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Table** | Spécifiez le nom de la table à utiliser. |
| **Action on table** | Sélectionnez une opération à effectuer sur la table définie, vous pouvez effectuer l'une des opérations suivantes :

• **Default** : aucune opération n’est effectuée.
• **Drop and create table** : la table est supprimée et créée à nouveau.
• **Create table** : la table n’existe pas et est créée.
• **Create table if does not exist** : la table est créée si elle n’existe pas.
• **Drop table if exist and create** : la table est supprimée si elle existe déjà et créée à nouveau.
• **Clear table** : le contenu de la table est supprimé. Vous pouvez annuler cette opération.
• **Truncate** : le contenu de la table est supprimé. |
| **Turn on identity insert** | Cochez cette case pour utiliser votre séquence personnelle pour la valeur de l’identité des enregistrements insérés (au lieu que SQL Server prenne la valeur séquentielle suivante). |
| **Action on data** | Sélectionnez une action à effectuer sur les données de la table définie.

• **Insert** : ajoute de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
• **Single Insert Query** : ajoute les entrées à la table en un lot.
• **Update** : met à jour les entrées existantes.
• **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée. |
• **Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas dans le pool d’index, un nouvel enregistrement est inséré.

• **Delete** : supprime les entrées correspondantes au flux d’entrée.

• **Insert if not exist** : ajoute de nouvelles entrées à la table si elles n’existent pas.

Il est nécessaire de spécifier au moins une colonne comme clé primaire sur laquelle baser les opérations **Update et Delete**. Pour ce faire, cliquez sur **Edit Schema** et cochez les cases à côté des colonnes que vous souhaitez définir comme clés primaires. Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings**, vue dans laquelle vous pouvez définir simultanément des clés primaires pour les opérations **Update et Delete**. Pour ce faire, cochez la case **Use field options** et, dans la colonne **Key in update column**, cochez les cases à côté des colonnes à utiliser comme base pour l’opération **Update**. Répétez l’opération dans la colonne **Key in delete column** pour l’opération **Delete**.

La fonctionnalité de schéma dynamique peut être utilisée dans les modes suivants : **Insert, Update, Insert or update, Update or insert et Delete**.

**Specify identity field**

Cochez cette case pour spécifier quel est le champ Identity, constitué d’un numéro d’identification incrémenté automatiquement.

Lorsque cette case est cochée, trois autres champs s’affichent :

• **Identity field** : sélectionnez dans la liste la colonne que vous souhaitez définir comme champ identity.

• **Start value** : saisissez une valeur de départ, utilisée pour la première ligne chargée dans la table.

• **Step** : saisissez une valeur d’incrément, ajoutée à la valeur de la ligne précédemment chargée.

Cette case est disponible si vous sélectionnez l’option **Drop and create table, Create table, Create table if not exists ou Drop table if exists and create** dans la liste **Action on table**. Elle ne s’affiche pas si vous sélectionnez l’option **Enable parallel execution** dans la vue **Advanced settings**. Si vous cochez cette case sans avoir cochée la case **Turn on identity insert** mais en ayant sélectionné **Create table if not exists** dans la liste **Action on table** et si la table spécifiée n’existe pas, seule une table est créée sans qu’aucune donnée y soit insérée.

Vous pouvez également spécifier le champ Identity à partir du schéma du composant. Pour ce faire, paramétrez le type de base de données (DB Type) de la colonne correspondante en **INT IDENTITY**.

Lorsque la case **Specify identity field** est cochée, le type de base de données (DB Type) **INT IDENTITY** du schéma est ignoré.
### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Specify a data source alias

Cochez cette case et, dans le champ **Data source alias** qui s’affiche, spécifiez l’alias d’une source de données créée du côté Talend Runtime, pour utiliser le pool de connexions partagées défini dans la configuration de la source de données. Cette option fonctionne uniquement lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour un scénario associé, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647.

Cette case est indisponible lorsque la case **Use an existing Connection** est cochée.

### Die on error

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

DÉCOchez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.

### Advanced settings

#### Additional JDBC Parameters

Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, `encrypt=true;trustServerCertificate=false;hostNameInCert`
<table>
<thead>
<tr>
<th>Commit every</th>
<th>Saisissez le nombre de lignes à terminer avant de commiter les lots de lignes dans la base de données. Cette option assure la qualité des transactions (mais pas le rollback) et de meilleures performances sur les exécutions.</th>
</tr>
</thead>
</table>
| Additional Columns | Cette option n’est pas disponible si vous créez (avec suppression ou sans) la table de base de données. Cette option vous permet d’appeler des fonctions SQL afin d’effectuer des actions sur des colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.  
  
  - **Name** : saisissez le nom de la colonne à modifier ou à insérer en tant que nouvelle colonne.  
  - **SQL expression** : saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.  
  - **Position** : sélectionnez Before, Replace ou After, en fonction de l’action à effectuer sur la colonne de référence.  
  - **Reference column** : saisissez une colonne de référence pouvant être utilisée pour placer ou remplacer la nouvelle colonne ou la colonne modifiée. |
| Use field options | Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données. |
| Ignore date out of range | Cochez cette case pour ignorer la validation de la date et insérer les données directement dans la base de données pour les types de données DATE, DATETIME, DATETIME2 et DATETIMEOFFSET. |
| Enable debug mode | Cochez cette case pour afficher chaque étape du traitement des entrées dans la base de données. |
| Support null in “SQL WHERE” statement | Cochez cette case pour prendre en compte les valeurs Null d’une table de base de données..  
  Assurez-vous que la case Nullable est bien cochée pour les colonnes du schéma correspondantes. |
| Use Batch | Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.  
  Cette case est disponible uniquement si vous avez choisi l’option Insert, Update, Single Insert Query ou Delete dans la liste Action on data.  
  Si vous sélectionnez l’option Single Insert Query dans la liste Action on data, la taille des lots doit être inférieure ou égale à la limite des marqueurs de paramètres autorisés par le pilote JDBC (généralement 2000), divisé par le nombre de colonnes. Pour plus d’informations |
**tSQLDWHOutput**

<table>
<thead>
<tr>
<th>Batch Size</th>
<th>Spécifiez le nombre d’enregistrements à traiter dans chaque lot. Ce champ est disponible uniquement lorsque la case <strong>Use batch mode</strong> est cochée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_UPDATED</td>
<td>Nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_INSERTED</td>
<td>Nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_DELETED</td>
<td>Nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_REJECTED</td>
<td>Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de fin dans un Job ou un sous-job et nécessite un lien d’entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue <strong>Basic settings</strong> devient inutilisable.</td>
</tr>
</tbody>
</table>
Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

**Limitation**

Lorsque l’option **Single Insert Query** est sélectionnée dans la liste **Action on data**, une Instruction Préparée SQL est générée, par exemple, `INSERT INTO table (col1, col2, col3) VALUES (?, ?, ?), (?, ?, ?), (?, ?, ?), (?, ?, ?)`. Entre les parenthèses se trouvent les groupes de paramètres, dont le nombre ne peut en général pas dépasser 2000, selon le pilote JBDC. La taille des lots doit être configurée de manière à respecter cette limite, c’est-à-dire qu’elle ne doit pas dépasser 2000.

 Certaines fonctionnalités supportées par d’autres bases de données ne sont pas supportées par Azure SQL Data Warehouse. Pour plus d’informations, consultez Fonctionnalités de table non prises en charge.

Notez que, lorsque vous créez ou supprimez une table avec ce composant, il est recommandé d’utiliser la fonctionnalité de commit automatique via la connexion à la base de données créée par un composant `tSQLDWHConnection` et de cocher la case **Auto Commit** dans la vue **Advanced settings** de ce `tSQLDWHConnection`, au lieu d’utiliser un composant `tSWLDWHCCommit`.

**Scénario associé**

Aucun scénario n’est disponible pour ce composant.
tSQLDWHRollback

Ce composant annule le commit d'une transaction dans la base de données Azure SQL Data Warehouse connectée, afin d'empêcher les commits partiels de transactions si une erreur survient.

**Propriétés du tSQLDWHRollback Standard**

Ces propriétés sont utilisées pour configurer le tSQLDWHRollback s'exécutant dans le framework de Jobs Standard.

Le composant tSQLDWHRollback Standard appartient à deux familles : Cloud et Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Component List</th>
<th>Sélectionnez dans la liste le composant tSQLDWHConnection si plus d'une connexion est utilisée dans le Job.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close Connection</td>
<td>Cette case est cochée par défaut. Cela vous permet de fermer la connexion à la base de données une fois l'opération terminée. Décrochez cette case pour continuer à utiliser la connexion sélectionnée, une fois la tâche effectuée par le composant.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant. |

**Global Variables**

| ERROR_MESSAGE | Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. |

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est généralement utilisé avec d'autres composants Azure SQL Data Warehouse, notamment le tSQLDWHConnection et le tSQLDWHCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez</td>
</tr>
</tbody>
</table>
dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Limitation

Certaines fonctionnalités supportées par d’autres bases de données ne sont pas supportées par Azure SQL Data Warehouse. Pour plus d’informations, consultez Fonctionnalités de table non prises en charge.

Scénario associé

Aucun scénario n’est disponible pour ce composant.
tSQLDWHRow

Ce composant exécute une requête SQL sur une base de données Azure SQL Data Warehouse.

**Propriétés du tSQLDWHRow Standard**

Ces propriétés sont utilisées pour configurer le tSQLDWHRow s’exécutant dans le framework de Jobs Standard.

Le composant tSQLDWHRow Standard appartient à deux familles : Cloud et Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, si vous devez partager une connexion existante entre ces deux niveaux, par exemple pour partager la connexion créée par le Job père au Job fils, vous devez :

1. au niveau du Job père, enregistrer la connexion à la base de données à partager dans la vue **Basic settings** du composant de connexion créant cette connexion à la base de données.

2. au niveau du Job fils, utiliser un composant de connexion dédié afin de lire cette connexion enregistrée.

Pour un exemple concernant le partage de connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

| Property Type | Sélectionnez la manière de configurer les informations de connexion.

- **Built-In** : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.


| JDBC Provider | Sélectionnez le fournisseur du pilote JDBC à utiliser.

<p>| Host | Spécifiez l’adresse IP ou le nom d’hôte de l’entrepôt Azure SQL Data Warehouse à utiliser. |</p>
<table>
<thead>
<tr>
<th>Port</th>
<th>Spécifiez le numéro du port d’écoute de l’entrepôt Azure SQL Data Warehouse à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Saisissez le nom du schéma Azure SQL Data Warehouse.</td>
</tr>
<tr>
<td>Database</td>
<td>Spécifiez le nom de l’entrepôt Azure SQL Data Warehouse à utiliser.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d’authentification utilisateur pour accéder à Azure SQL Data Warehouse. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
  - Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
Cliquez sur Edit schema pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.  
  - View schema : sélectionnez cette option afin de voir le schéma.  
  - Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
  - Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| Table name | Spécifiez le nom de la table à utiliser. |
| Turn on identity insert | Cochez cette case pour utiliser votre séquence personnelle pour la valeur de l’identité des enregistrements insérés (au lieu que SQL Server prenne la valeur séquentielle suivante). |
| Query Type | Sélectionnez la manière de configurer la requête. |
- **Built-in**: saisissez manuellement l'instruction de requête ou construisez-la graphiquement à l’aide du SQLBuilder.
- **Repository**: Sélectionnez la requête correspondant, stockée dans le Repository. Le champ Query est renseigné.

<table>
<thead>
<tr>
<th><strong>Guess Query</strong></th>
<th>Cliquez sur ce bouton pour générer la requête correspondant au schéma de votre table, dans le champ Query.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query</strong></td>
<td>Spécifiez la requête en faisant particulièrement attention à mettre les champs dans un ordre correct, afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td><strong>Specify a data source alias</strong></td>
<td>Cochez cette case et, dans le champ Data source alias qui s’affiche, spécifiez l’alias d’une source de données créée du côté Talend Runtime, pour utiliser le pool de connexions partagées défini dans la configuration de la source de données. Cette option fonctionne uniquement lorsque vous déployez et exécutez votre Job dans Talend Runtime. Pour un scénario associé, consultez Scénario : Déploiement de votre Job dans Talend Runtime pour récupérer les données d’une base de données MySQL à la page 2647. Cette case est indisponible lorsque la case Use an existing Connection est cochée.</td>
</tr>
<tr>
<td><strong>Die on error</strong></td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque des erreurs sont ignorées, vous pouvez collecter les lignes en erreur à l’aide d’un lien Row &gt; Reject.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>Additional JDBC Parameters</strong></th>
<th>Définissez des propriétés de connexion supplémentaires pour la connexion à la base de données que vous créez. Les propriétés sont séparées par un point-virgule et chaque propriété est une paire clé-valeur. Par exemple, encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30; pour une connexion à la base de données Azure SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Propagate QUERY’s recordset</strong></td>
<td>Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column. Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type Object. Ce composant est généralement suivi d’un tParseRecordSet.</td>
</tr>
</tbody>
</table>
Use PreparedStatement

Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau Set PreparedStatement Parameter, définissez les valeurs des paramètres représentés par des "?" dans l'instruction SQL définie dans le champ Query de l’onglet Basic settings.

- **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL.
- **Parameter Type** : Saisissez le type du paramètre.
- **Parameter Value** : Saisissez la valeur du paramètre.

Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

Commit every

Saisissez le nombre de lignes à terminer avant de commiter les lots de lignes dans la base de données. Cette option assure la qualité des transactions (mais pas le rollback) et de meilleures performances sur les exécutions.

tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

Global Variables

<table>
<thead>
<tr>
<th>ERROR_MESSAGE</th>
<th>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUERY</td>
<td>Instruction de requête en cours de traitement. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée.
## Scénario associé

Aucun scénario n’est disponible pour ce composant.
**tSQLiteClose**

Ce composant permet de fermer une connexion à la base de données SQLite.

**Propriétés du tSQLiteClose Standard**

Ces propriétés sont utilisées pour configurer le tSQLiteClose s’exécutant dans le framework de Jobs Standard.

Le composant tSQLiteClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant <strong>tSQLiteConnection</strong> dans la liste.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tSQLiteCommit

Ce composant utilise une connexion unique pour commiter en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

Le tSQLiteCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tSQLiteCommit Standard

Ces propriétés sont utilisées pour configurer le tSQLiteCommit s’exécutant dans le framework de Jobs Standard.

Le composant tSQLiteCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tSQLiteConnection dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. Avertissement : Si vous utilisez un lien de type Row &gt; Main pour relier le tSQLiteCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant est généralement utilisé avec des composants SQLite, notamment les composants tSQLiteConnection et tSQLiteRollback.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic settings</th>
</tr>
</thead>
</table>
| Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. 

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


## Scénario associé

Pour plus d’informations relatives au fonctionnement du composant tSQLiteCommit, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tSQLiteConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

**Propriétés du tSQLiteConnection Standard**

Ces propriétés sont utilisées pour configurer le tSQLiteConnection s’exécutant dans le framework de Jobs Standard.

Le composant tSQLiteConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Table Schema</td>
<td>Nom du schéma.</td>
</tr>
<tr>
<td>Use or register a shared DB Connection</td>
<td>Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ Shared DB Connection Name qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents. Cette option est incompatible avec les options Use dynamic job et Use an independent process to run subjob du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.</td>
</tr>
</tbody>
</table>


## Advanced settings

| Auto Commit | Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.  
Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.  
Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commitne pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log, aussi bien au niveau du Job qu’au niveau de chaque composant. |

## Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec des composants SQLite, notamment les composants tSQLiteCommit et tSQLiteRollback. |

## Scénarios associés

Pour un scénario associé au composant tSQLiteConnection, consultez tMysqlConnection à la page 2618.
tSQLiteInput

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

Le tSQLiteInput lit une base de données et en extrait des champs à l’aide de requêtes. Un petit moteur de base de données SQLite lui est intégré, il n’a donc pas besoin d’être connecté à un serveur de base de données.

Propriétés du tSQLiteInput Standard

Ces propriétés sont utilisées pour configurer le tSQLiteInput s’exécutant dans le framework de Jobs Standard.

Le composant tSQLiteInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existante entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Database</th>
<th>Chemin d’accès au fichier de la base de données SQLite.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <em>line</em> lors du nommage des champs.</td>
</tr>
</tbody>
</table>

- **Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

| Query type | La requête peut être construite manuellement (Built-in) pour un Job particulier ou stockée dans le Repository s’il s’agit d’une requête fréquemment utilisée, pour en faciliter l’utilisation. |
**Query**

| Query | Si elle n’est pas stockée dans le **Repository**, saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma. |

---

**Advanced settings**

<table>
<thead>
<tr>
<th>Trim all the String/Char columns</th>
<th>Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim column</td>
<td>Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td>tStat Catcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

---

**Global Variables**

| Global Variables | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.  
**QUERY** : requête traitée. Cette variable est une variable **Flow** et retourne une chaîne de caractères.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**. |

---

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est tout à fait autonome du fait de son moteur de base de données SQLite intégré. Il s’agit d’un composant de début pouvant initier un traitement de flux de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez</td>
</tr>
</tbody>
</table>
Scénario : Filtrer des données SQLite

Le scénario suivant décrit un Job assez simple qui permet d’appliquer une requête donnée utilisant un filtre pour extraire des lignes à partir d’une base de données SQLite source et alimenter une table SQLite en sortie.

- Cliquez-déposez à partir de la Palette les composants tSQLiteInput et tSQLiteOutput.
- Reliez le composant d’entrée et de sortie à l’aide d’un lien de type row Main.
- Dans l’onglet Basic settings du composant tSQLiteInput, saisissez le chemin d’accès ou parcourez vos dossiers jusqu’au fichier d’entrée à la base de donnée SQLite.
• Le fichier contient des centaines de lignes et comprend une colonne `ip` servant de base à la requête.

• Dans l’onglet **Basic settings** du composant **tSQLiteInput**, modifiez le schéma afin qu’il corresponde à la structure de la table.

  - **Property Type**: `Built-In`
  - **Use an existing connection**: désactivé
  - **Database**: `C:/Input/Talend_rb/SQLite/ipcountry.db`
  - **Schema**: `Built-In`
  - **Table Name**: `***`
  - **Query Type**: `Built-In`
  - **Query**: 
    ```
    "select * from download where ip = 1195650472"
    ```

• Dans le champ **Query**, renseignez la requête choisie par rapport à la colonne `ip`.

• Sélectionnez les paramètres d’encodage correspondants.

• Pour le composant **tSQLiteOutput**, dans le champ **Database** de l’onglet **Basic settings**, sélectionnez le chemin d’accès à la base de données de sortie.
Dans le champ **Table**, renseignez le nom de la table à alimenter avec les données sélectionnées.

A partir des listes déroulantes **Action on table** et **Action on Data**, sélectionnez les actions à effectuer sur la table et sur les données. Dans le cas présent, les actions sont, respectivement, *Drop and create* et *Insert*.

Synchronisez le schéma avec le schéma d’entrée.

Sélectionnez l’encodage puis définissez la limite de commit.

Enregistrez le Job puis exécutez-le en appuyant sur **F6**.

<table>
<thead>
<tr>
<th>id</th>
<th>version</th>
<th>download_date</th>
<th>ip</th>
<th>type</th>
<th>type_os</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20028</td>
<td>TOS-Win32-20061011</td>
<td>13/11/2006</td>
<td>1195650472</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>20031</td>
<td>TOS-Win32-20061011</td>
<td>13/11/2006</td>
<td>1195650472</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>20032</td>
<td>TOS-Win32-20061011</td>
<td>13/11/2006</td>
<td>1195650472</td>
<td>1</td>
</tr>
</tbody>
</table>

Les données sélectionnées sont renvoyées vers le fichier SQLite défini.
Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Le tSQLiteOutput écrit, met à jour, modifie ou supprime les données d’une base de données. Un petit moteur de base de données SQLite lui est intégré, il n’a donc pas besoin d’être connecté à un serveur de base de données.

**Propriétés du tSQLiteOutput Standard**

Ces propriétés sont utilisées pour configurer le tSQLiteOutput s’exécutant dans le framework de Jobs Standard.

Le composant tSQLiteOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque** :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Use an existing connection**

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

**Remarque** :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

Cliquez sur cette icône pour ouvrir l'assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.

Pour plus d'informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Database</th>
<th>Chemin d'accès au fichier de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu'une seule table à la fois.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l'une des opérations suivantes sur les données de la table sélectionnée :

- **None** : n'effectuer aucune opération de table.
- **Drop and create the table** : supprimer la table puis en créer une nouvelle.
- **Create a table** : créer une table qui n'existe pas encore.
- **Create table if doesn't exist** : créer la table si nécessaire.
- **Clear a table** : supprimer le contenu de la table.
- **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| Action on data | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

- **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.
- **Update** : Mettre à jour les entrées existantes.
- **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Update or insert** : met à jour l'enregistrement avec la référence donnée. Si l'enregistrement n'existe pas, un nouvel enregistrement est inséré.
- **Delete** : Supprimer les entrées correspondantes au flux d'entrée. |

**Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations Update et Delete. Pour cela, cliquez sur le bouton [...] à côté du champ Edit Schema et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous
souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet Advanced settings pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case Use field options et sélectionnez la case Key in update correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases Key in delete pour les opérations de suppression (Delete).

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Die on error

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en
erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Commit every</strong></td>
<td>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</td>
</tr>
<tr>
<td><strong>Additional Columns</strong></td>
<td>Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Saisissez le nom de la colonne à modifier ou à insérer.</td>
</tr>
<tr>
<td><strong>SQL expression</strong></td>
<td>Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.</td>
</tr>
<tr>
<td><strong>Position</strong></td>
<td>Sélectionnez <strong>Before</strong>, <strong>Replace</strong> ou <strong>After</strong>, en fonction de l’action à effectuer sur la colonne de référence.</td>
</tr>
<tr>
<td><strong>Reference column</strong></td>
<td>Saisissez une colonne de référence que le composant <strong>tSQLiteOutput</strong> peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.</td>
</tr>
<tr>
<td><strong>Use field options</strong></td>
<td>Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.</td>
</tr>
<tr>
<td><strong>Enable debug mode</strong></td>
<td>Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.</td>
</tr>
<tr>
<td><strong>Use Batch</strong></td>
<td>Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.</td>
</tr>
<tr>
<td><strong>Remark</strong></td>
<td>Cette case est disponible lorsque vous sélectionnez <strong>Insert</strong>, <strong>Update</strong>, ou <strong>Delete</strong> dans la liste <strong>Action on data</strong>.</td>
</tr>
<tr>
<td><strong>Batch Size</strong></td>
<td>Spéifiéz le nombre d’enregistrements à traiter dans chaque lot.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>
Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong> : nombre de lignes traitées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong> : nombre de lignes mises à jour. Cette variable est une variable <em>After</em> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong> : nombre de lignes insérées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_DELETED</strong> : nombre de lignes supprimées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>NB_LINE_REJECTED</strong> : nombre de lignes rejetées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
<td></td>
</tr>
<tr>
<td><strong>QUERY</strong> : requête traitée. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères.</td>
<td></td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option.</td>
<td></td>
</tr>
</tbody>
</table>

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant doit absolument être relié à un composant d’entrée (Input). Il permet de faire des actions sur une table ou les données d’une table d’une base de données SQLite. Il permet aussi de créer un flux de rejet avec un lien Row &gt; Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic settings</th>
<th></th>
</tr>
</thead>
</table>
| Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exem
ple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

Pour un scénario associé au composant **tSQLiteOutput**, consultez **tSQLiteInput** à la page 3858.
**tSQLiteRollback**

Ce composant annule la transaction commitée dans la base de données SQLite connectée.

**Propriétés du tSQLiteRollback Standard**

Ces propriétés sont utilisées pour configurer le tSQLiteRollback s’exécutant dans le framework de Jobs Standard.

Le composant tSQLiteRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant de connexion <strong>tSQLiteConnection</strong> dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en association avec des composants SQLite, notamment avec le <strong>tSQLiteCommit</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [*] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez...</td>
</tr>
</tbody>
</table>
dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénario associé

Pour un scénario associé au tSQLiteRollback, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623 du composant tMysqlRollback.
tSQLiteRow

Ce composant exécute une requête définie dans une base de données spécifiée et utilise les paramètres de la colonne. Prédéfinie utilise le flux.

Une instruction préparée utilise le flux d’entrée pour remplacer les paramètres substituables par les valeurs définies pour chaque paramètre. Ce composant est très utile quand il s’agit de faire des mises à jour.

Propriétés du tSQLiteRow Standard

Ces propriétés sont utilisées pour configurer le tSQLiteRow s’exécutant dans le framework de Jobs Standard.

Le composant tSQLiteRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
</table>
| Property type | Peut être Built-in ou Repository

**Built-in** : Propriétés utilisées ponctuellement.

**Repository** : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.

| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

### Database

Chemin d'accès au fichier de la base de données SQLite.

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-in**: Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.
- **Repository**: Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Table Name

Nom de la table à lire.

**Query type**

Peut être **Built-in** ou **Repository**.

- **Built-in**: Saisissez manuellement votre requête ou construisez-la à l'aide de SQLBuilder.
- **Repository**: Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.

**Query**

Saisissez votre requête en faisant particulièrement attention à l'ordre des champs afin qu'ils correspondent à la définition du schéma.
### Advanced settings

#### Propagate QUERY’s recordset

Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste `use column`.

**Remarque :**

Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type `Object`. Ce composant est généralement suivi du `tParseRecordSet`.

#### Use PreparedStatement


- **Parameter Index** : Saisissez la position du paramètre dans l'instruction SQL.
- **Parameter Type** : Saisissez le type du paramètre.
- **Parameter Value** : Saisissez la valeur du paramètre.

**Remarque :**

Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

#### Commit every

Nombre de lignes avant le commit.

#### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**QUERY** : requête traitée. Cette variable est une variable `Flow` et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable `After` et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case `Die on error` est décochée, si le composant a cette option.

Une variable `Flow` fonctionne durant l’exécution d’un composant. Une variable `After` fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

## Scénario : Mettre à jour des lignes SQLite

Le scénario suivant décrit un Job qui permet de mettre à jour un fichier de base de données SQLite en utilisant une requête prédéfinie et un fichier délimité.

- Cliquez-déposez les composants `tFileInputDelimited` et `tSQLiteRow`.

![Diagram](image)
Dans l'onglet **Basic settings** du composant **tFileInputDelimited**, parcourez les dossiers et sélectionnez le fichier d'entrée à utiliser pour mettre à jour les lignes dans la base de données.

- Il n'y a ni en-tête (**Header**) ni pied de page (**Footer**). Le séparateur de lignes (**Row separator**) est un retour chariot et le séparateur de champs (**Field separator**) un point-virgule.

- Modifiez le schéma s'il n'est pas stocké dans le **Repository**.

- Assurez-vous que la longueur et le type définissent correctement les colonnes.

Dans l'espace de modélisation graphique, double-cliquez sur le composant **tSQLiteRow** pour afficher sa vue **Component**.

- Dans l'onglet **Basic settings** du composant **tSQLiteRow**, renseignez le champ **Database** avec le chemin d'accès au fichier à mettre à jour.
• Le schéma est en mode read-only puisqu’il doit correspondre au schéma d’entrée.

• Saisissez la requête ou récupérez-la à partir du Repository. Dans le cas présent, la colonne type_os a été mise à jour en fonction de la valeur id définie dans le flux entrant. La requête se présente comme suit :

"Update download set type_os=? where id=?"

• Dans l’onglet Advanced settings, cochez la case Use PreparedStatement pour afficher la table des paramètres substituables.

La table download de la base de données SQLite est ainsi mise à jour avec le nouveau code type_os, selon le fichier délimité d’entrée.

Scénario associé

Pour un scénario associé, consultez :

• Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
**tSQLTemplate**

Ce composant exécute simplement des actions sur une base de données ou des modèles d'instructions SQL personnalisés, par exemple pour supprimer ou créer une table.

Le tSQLTemplate fournit un ensemble de modèles d'instructions SQL pour un certain nombre de SGBD, afin de faciliter les actions les plus communes sur une base de données. Ces modèles personnalisés sont accessibles via la vue **SQL Template**. De plus, il est possible de personnaliser les modèles d'instructions SQL selon vos besoins.

**Propriétés du tSQLTemplate Standard**

Ces propriétés sont utilisées pour configurer le tSQLTemplate s'exécutant dans le framework de Jobs Standard.

Le composant tSQLTemplate Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Type</td>
<td>Sélectionnez dans la liste le type de base de données à laquelle vous connecter.</td>
</tr>
<tr>
<td>Component List</td>
<td>Sélectionnez dans la liste le composant de connexion à la base de données à utiliser, si plus d’une connexion est utilisée dans le Job.</td>
</tr>
<tr>
<td>Database name</td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td>Table name</td>
<td>Saisissez le nom de la table sur laquelle exécuter les modèles d'instructions SQL.</td>
</tr>
</tbody>
</table>
| Schema et Edit schema   | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :
  - **View schema** : sélectionnez cette option afin de voir le schéma.
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
**Built-in** : Le schéma sera créé et conservé pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.


### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log aussi bien au niveau du Job qu'au niveau des composants. |

### Global Variables

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.  
**QUERY** : requête traitée. Cette variable est une variable *Flow* et retourne une chaîne de caractères.  
**ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  
Une variable *Flow* fonctionne durant l'exécution d'un composant. Une variable *After* fonctionne après l'exécution d'un composant.  
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d'informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

### Utilisation

| Règle d'utilisation | Ce composant est un composant de début. Il doit être utilisé avec d'autres composants de bases de données, notamment des composants de connexion et de commit. |

| SQL Template | SQL Template List |

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion HDFS parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à des fichiers dans différents systèmes HDFS ou dans différentes distributions, en particulier lorsque vous |
travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté dans un *Studio Talend* indépendant.

La table *Dynamic settings* est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue *Basic settings*. Lorsqu’un paramètre dynamique est configuré, la liste *Component List* de la vue *Basic settings* devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez *Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte* à la page 2641 et *Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement* à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

---

**Scénario associé**

Pour un scénario associé, consultez *Scénario : Filtre et agréger les colonnes d’une table directement dans le SGBD* à la page 3883.
tSQLTemplateAggregate

Ce composant fournit un ensemble de métriques basées sur des valeurs ou des calculs.

Le tSQLTemplateAggregate collecte les données d’une ou plusieurs colonnes et gère ces données en un seul bloc. Ce composant exécute en temps réel des transformations de données dans le SGBD lui-même.

Propriétés du tSQLTemplateAggregate Standard

Ces propriétés sont utilisées pour configurer le tSQLTemplateAggregate s’exécutant dans le framework de Jobs Standard.

Le composant tSQLTemplateAggregate Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Type</td>
<td>Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.</td>
</tr>
<tr>
<td>Component List</td>
<td>Sélectionnez le composant de connexion à la base de données dans la liste si vous utilisez plus d’une connexion dans le Job en cours.</td>
</tr>
<tr>
<td>Database name</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Source table name</td>
<td>Nom de la table contenant les données dont vous souhaitez collecter les données.</td>
</tr>
<tr>
<td>Target table name</td>
<td>Nom de la table dans laquelle vous souhaitez écrire les données collectées et transformées.</td>
</tr>
</tbody>
</table>
| Schema et Edit schema     | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
  • View schema : sélectionnez cette option afin de voir le schéma.
  • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
  • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la |
<table>
<thead>
<tr>
<th>Métadonnée du schéma dans la fenêtre</th>
<th>[Repository Content].</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>Operations</strong></td>
<td>Sélectionnez le type d'opération et la valeur à utiliser pour le calcul et le champ de sortie.</td>
</tr>
<tr>
<td><strong>Output Column</strong></td>
<td>Sélectionnez le champ de destination dans la liste.</td>
</tr>
<tr>
<td><strong>Function</strong></td>
<td>Sélectionnez une des opérations suivantes à effectuer sur les données :</td>
</tr>
<tr>
<td>count</td>
<td>calcule le nombre de lignes,</td>
</tr>
<tr>
<td>min</td>
<td>sélectionne la plus petite valeur,</td>
</tr>
<tr>
<td>max</td>
<td>sélectionne la plus grande valeur,</td>
</tr>
<tr>
<td>avg</td>
<td>calcule la moyenne,</td>
</tr>
<tr>
<td>sum</td>
<td>calcule la somme,</td>
</tr>
<tr>
<td>count (distinct)</td>
<td>compte le nombre de lignes sans les doublons.</td>
</tr>
<tr>
<td><strong>Input column position</strong></td>
<td>Sélectionnez la colonne d'entrée à partir de laquelle vous souhaitez collecter les données à agréger.</td>
</tr>
<tr>
<td><strong>Group by</strong></td>
<td>Définit les ensembles d'agrégation, dont les valeurs sont utilisées pour les calculs.</td>
</tr>
<tr>
<td><strong>Output Column</strong></td>
<td>Sélectionnez le libellé de colonne dans la liste fournie, basée sur la structure de schéma que vous avez définie. Vous pouvez ajouter autant de colonnes de sortie que vous le souhaitez afin d'affiner les agrégations.</td>
</tr>
<tr>
<td><strong>Input Column position</strong></td>
<td>Faites la correspondance entre les libellés des colonnes d'entrée avec ceux des colonnes de sortie, dans le cas où vous souhaitez que les libellés du schéma de sortie soient différents du schéma d'entrée.</td>
</tr>
<tr>
<td><strong>Advanced settings</strong></td>
<td></td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>
Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>QUERY</strong></td>
<td>requête traitée. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable <em>Flow</em> fonctionne durant l’exécution d’un composant. Une variable <em>After</em> fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SQL Template</strong></td>
<td>SQLTemplate List</td>
</tr>
</tbody>
</table>

Scénario : Filtre et agrége les colonnes d’une table directement dans le SGBD

Le scénario suivant crée un Job qui ouvre une connexion à une base de données Mysql et :

- crée un schéma à partir de la table d’une base de données, les lignes de ce schéma correspondent aux noms des colonnes spécifiées dans le filtre,
- filtre les colonnes de cette même table afin de n’obtenir que les données correspondant aux colonnes filtrées,
- collecte les données de la colonne filtrée, regroupées en fonction d’une valeur spécifique et écrit les données agrégées dans la table de cible.

Pour filtrer et agréger les colonnes de la table de votre base de données :

- Cliquez et déposez les composants suivants dans l’espace de modélisation : tELTMysqlconnection, tSQLTemplateFilterColumns, tSQLTemplateFilterRows, tSQLTemplateAggregate, tSQLTemplateCommit et tSQLTemplateRollback.
- Connectez les cinq composants à l’aide de liens de type OnComponentOk.
• Connectez les composants **tSQLTemplateAggregate** et **tSQLTemplateRollback** à l'aide d'un lien de type **OnComponentError**.

• Dans l'espace de modélisation, sélectionnez le **tMysqlConnection** et cliquez sur la vue **Component** pour paramétrer ses propriétés.

• Dans l'onglet **Basic settings**, renseignez manuellement les informations de connexion ou sélectionnez-les à partir des variables de contexte, via la raccourci **Ctrl+Espace** cliquez sur le champ correspondant si les vous avez stocké sous le nœud **DB connection** du répertoire **Metadata** du **Repository**.

Pour plus d’informations sur les métadonnées, consultez le **Guide utilisateur du Studio Talend**.

• Dans l'espace de modélisation, sélectionnez le composant **tSQLTemplateFilterColumns** et cliquez sur la vue **Component** pour paramétrer ses propriétés.
• Dans la liste déroulante **Database type**, sélectionnez la base de données adéquate.

• Dans la liste déroulante **Component list**, sélectionnez le composant de connexion à la base de données si plusieurs composants de connexion sont utilisés dans votre Job.

• Dans le champ **Database name**, saisissez le nom de la base de données à laquelle vous vous êtes connecté.

• Dans le champ **Source table name**, saisissez le nom de votre table source contenant les données que vous souhaitez traiter et cliquez sur le bouton [...] à côté du champ **Edit schema** pour définir la structure des données de votre table source.

• Dans le champ **Target table name**, saisissez le nom de la table cible contenant vos nouvelles données traitées et cliquez sur le bouton [...] à côté du champ **Edit schema** pour définir la structure de la table cible.

**Remarque :**

Lorsque vous définissez la structure des données de la table source, le nom des colonnes apparaît automatiquement dans la colonne **Column** du tableau **Column filters**.

Dans ce scénario, la table source contient les cinq colonnes : *id, First_Name, Last_Name, Address* et *id_State*.

• Dans le tableau **Column filters**, filtrez les colonnes en cochant la case des colonnes que vous souhaitez écrire dans votre table cible.

Dans ce scénario, le composant **tSQLTemplateFilterColumns** ne garde que les colonnes *id, First_Name* et *id_State* de la table source.

**Remarque :**

Dans la vue **Component**, vous pouvez cliquer sur l’onglet **SQL Template** et ajouter des modèles SQL du système ou créer vos propres modèles SQL et les utiliser dans votre Job afin d’effectuer les opérations que vous souhaitez. Pour plus d’informations, consultez Propriétés du **tSQLTemplateFilterColumns Standard** à la page 3890.

• Dans l’espace de modélisation, sélectionnez le composant **tSQLTemplateFilterRows** et cliquez sur l’onglet **Component** pour paramétrer ses propriétés.
Dans le champ **Source table name**, saisissez le nom de votre table source contenant les données que vous souhaitez traiter et cliquez sur le bouton […] à côté du champ **Edit schema** pour définir la structure des données de votre table source.

Dans le champ **Target table name**, saisissez le nom de la table cible contenant vos nouvelles données traitées et cliquez sur le bouton […] à côté du champ **Edit schema** pour définir la structure de la table cible.

Dans ce scénario, la table source et la table cible contiennent les trois colonnes filtrées **id, First_Name et id_State**.

Dans le champ **Where condition**, saisissez la commande WHERE vous permettant de n’extraire que les champs correspondant à vos critères.

Dans ce scénario, le composant **tSQLTemplateFilterRows** filtre la colonne **First_Name** de la table source pour n’obtenir que les prénoms commencent par la lettre "a".

Dans l’espace de modélisation, sélectionnez le composant **tSQLTemplateAggregate** et cliquez sur la vue **Component** pour paramétrer ses propriétés.

Dans la liste déroulante **Database type**, sélectionnez la base de données adéquate.

Dans la liste **Component list**, sélectionnez le composant de connexion à la base de données si plusieurs composants de connexion sont utilisés dans votre Job.

Dans le champ **Database name**, saisissez le nom de la base de données à laquelle vous vous êtes connecté.

Dans le champ **Source table name**, saisissez le nom de votre table source contenant les données que vous souhaitez traiter et cliquez sur le bouton […] à côté du champ **Edit schema** pour définir la structure des données de votre table source.

Dans le champ **Target table name**, saisissez le nom de la table cible contenant vos nouvelles données traitées et cliquez sur le bouton […] à côté du champ **Edit schema** pour définir la structure de la table cible.

Le schéma de la table source est composé de trois colonnes : **id, First_Name et id_State**. Le schéma de la table cible est composé de deux colonnes : **customers_status et customers_number**. Dans ce scénario, vous souhaitez regrouper les clients par statut marital et compter le nombre de client pour chaque groupe marital. Pour cela, renseignez les tableaux **Operations et Group by**.
Dans le tableau **Operations**, cliquez sur le bouton 
[*] pour ajouter une ou plusieurs lignes et cliquez dans la cellule de la colonne **Output column** et sélectionnez la colonne de sortie qui contiendra les données à compter.

Cliquez dans la cellule de la colonne **Function** et sélectionnez l'opération à effectuer.

Dans le tableau **Group by**, cliquez sur le bouton 
[*] pour ajouter une ou plusieurs lignes et cliquez dans la cellule de la colonne **Output column** pour sélectionner la colonne de sortie qui contiendra les données agrégées.

Dans l'espace de modélisation, sélectionnez le composant **tSQLTemplateCommit** et cliquez sur la vue **Component** pour paramétrer ses propriétés.

Dans la liste déroulante **Database type**, sélectionnez la base de données adéquate.

Dans la liste déroulante **Component list**, sélectionnez le composant de connexion à la base de données si plusieurs composants de connexion sont utilisés dans votre Job.

Effectuez la même procédure pour le composant **tSQLTemplateRollback**.

Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

La table *aggregate_customers* composée de deux colonnes est créée dans la base de données. Elle regroupe les clients en fonction de leur statut marital et compte le nombre de client pour chaque statut marital.
tSQLTemplateCommit

Ce composant commite en une seule fois une transaction globale, en utilisant une connexion unique, au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

Le tSQLTemplateCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tSQLTemplateCommit Standard

Ces propriétés sont utilisées pour configurer le tSQLTemplateCommit s’exécutant dans le framework de Jobs Standard.

Le composant tSQLTemplateCommit Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Type</td>
<td>Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.</td>
</tr>
<tr>
<td>Component List</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant de connexion à la base de données que vous utilisez dans la liste.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>QUERY</td>
<td>requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
</tbody>
</table>
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>SQL Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant est généralement utilisé avec des composants ELT et notamment avec le composant <strong>tSQLTemplateRollback</strong>, ainsi qu’avec le composant ELT de connexion à la base de données que vous utilisez.</td>
<td>SQLTemplate List</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour plus d’informations relatives au fonctionnement du composant **tSQLTemplateCommit**, consultez *Scénario : Filtre et agréger les colonnes d’une table directement dans le SGBD* à la page 3883.
tSQLTemplateFilterColumns

Ce composant homogénéise des schémas en réorganisant ou en supprimant des colonnes ou en ajoutant de nouvelles colonnes.

Le tSQLTemplateFilterColumns modifie le schéma de la table d’une base de données en filtrant les colonnes. Ce composant filtre en temps réel les données dans le SGBD lui-même.

Propriétés du tSQLTemplateFilterColumns Standard

Ces propriétés sont utilisées pour configurer le tSQLTemplateFilterColumns s’exécutant dans le framework de Jobs Standard.

Le composant tSQLTemplateFilterColumns Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Type</td>
<td>Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.</td>
</tr>
<tr>
<td>Component List</td>
<td>Sélectionnez le composant de connexion à la base de données si vous utilisez plus d’une connexion dans votre Job en cours.</td>
</tr>
<tr>
<td>Database name</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Source table name</td>
<td>Nom de la table contenant les données à filtrer.</td>
</tr>
<tr>
<td>Target table name</td>
<td>Nom de la table dans laquelle vous souhaitez écrire les données filtrées.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :&lt;ul&gt;&lt;li&gt;View schema : sélectionnez cette option afin de voir le schéma.&lt;/li&gt;&lt;li&gt;Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.&lt;/li&gt;&lt;li&gt;Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content.&lt;/li&gt;&lt;/ul&gt;</td>
</tr>
</tbody>
</table>
tSQLTemplateFilterColumns


**Column filters**
Dans le tableau, cochez la case Filter pour filtrer toutes vos colonnes, ou, pour en filtrer moins, cliquez sur la ou les case(s) en face du nom de la ou des colonne(s).

**Advanced settings**

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

**NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.

**QUERY** : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

**Règle d’utilisation**
Ce composant est un composant intermédiaire. Vous pouvez l’utiliser avec d’autres composants de base de données, notamment les composants de connexion et de commit.

**SQL Template**
SQLTemplate List
Scénario associé

Pour un scénario associé, consultez Scénario : Filtre et agréger les colonnes d'une table directement dans le SGBD à la page 3883.
tSQLTemplateFilterRows

Configure des filtres de lignes sur n’importe quelle donnée source grâce à une commande WHERE.

Le tSQLTemplateFilterRows vous permet de filtrer les lignes d’une table. Ce composant filtre en temps réel les données dans le SGBD lui-même.

Propriétés du tSQLTemplateFilterRows Standard

Ces propriétés sont utilisées pour configurer le tSQLTemplateFilterRows s’exécutant dans le framework de Jobs Standard.

Le composant tSQLTemplateFilterRows Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Database Type</th>
<th>Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component List</td>
<td>Sélectionnez le composant de connexion à la base de données si vous utilisez plus d’une connexion dans le Job en cours.</td>
</tr>
<tr>
<td>Database name</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Source table name</td>
<td>Nom de la table contenant les données à filtrer.</td>
</tr>
<tr>
<td>Target table name</td>
<td>Nom de la table dans laquelle vous souhaitez écrire les données filtrées.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
</tbody>
</table>
### Built-in
Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository

### Where condition
Utilisez une commande WHERE pour définir un critère à partir duquel vous souhaitez filtrer vos données.

Vous pouvez utiliser la commande WHERE pour sélectionner des lignes spécifiques de votre table en fonction de certains critères ou certaines conditions.

### Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour collecter les données de log aussi bien au niveau du Job qu’au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant intermédiaire. Vous pouvez l’utiliser avec d’autres composants de base de données, notamment les composants de connexion et de commit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Template</td>
<td>SQLTemplate List</td>
</tr>
</tbody>
</table>
Scénario associé

Pour un scénario associé, consultez Scénario : Filtre et agréger les colonnes d’une table directement dans le SGBD à la page 3883.
tSQLTemplateMerge

Ce composant fusionne des données directement dans une table d’une base de données du SGBD. Pour cela, il crée et exécute une instruction MERGE.

Le tSQLTemplateMerge crée une instruction SQL MERGE sur les données dans une table de la base de données.

Propriétés du tSQLTemplateMerge Standard

Ces propriétés sont utilisées pour configurer le tSQLTemplateMerge s’exécutant dans le framework de Jobs Standard.

Le composant tSQLTemplateMerge Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

| Database Type | Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.
| Component list | Sélectionnez le composant ELT de connexion le plus pertinent dans la liste si vous prévoyez d’ajouter plus d’une connexion à votre Job en cours.
| Source table name | Nom de la table contenant les données sur lesquelles vous souhaitez effectuer les opérations d’insertion ou de mise à jour dans la table cible.
| Target table name | Nom de la table dans laquelle vous souhaitez effectuer les opérations d’insertion ou de mise à jour.
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
  - View schema : sélectionnez cette option afin de voir le schéma.
  - Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
  - Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propaguer ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].


| **Built-in** | Le schéma sera créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| **Merge ON** | Spécifiez les colonnes source et cible que vous souhaitez utiliser comme clés primaires. |
| **Use UPDATE (WHEN MATCHED)** | Cochez cette case pour mettre à jour les enregistrements existants. Une fois cette case cochée, la table **UPDATE Columns** apparaît, vous permettant de définir les colonnes dans lesquelles les enregistrements seront mis à jour. |
| **Specify additional output columns** | Cochez cette case pour mettre à jour les enregistrements dans des colonnes autres que celles listées dans la table **UPDATE Columns**. Une fois cette case cochée, la table **Additional UPDATE Columns** apparaît, vous permettant de définir des colonnes supplémentaires. |
| **Specify UPDATE WHERE clause** | Cochez cette case et saisissez une clause de type WHERE dans le champ **WHERE clause** afin de filtrer les données lors de l'opération de mise à jour.  

**Remarque :** Il est probable que cette option ne fonctionne pas avec certaines versions des bases de données, notamment Oracle 9i. |
| **Use INSERT (WHEN MATCHED)** | Cochez cette case pour insérer de nouveaux enregistrements. Une fois cette case cochée, la table **INSERT Columns** apparaît, vous permettant de définir des colonnes à impliquer dans l’opération d’insertion. |
| **Specify additional output columns** | Cochez cette case pour insérer des enregistrements dans les colonnes des colonnes autres que celles listées dans la table **INSERT Columns**. Une fois cette case cochée, la table **Additional INSERT Columns** apparaît, vous permettant de définir des colonnes supplémentaires. |
| **Specify INSERT WHERE clause** | Cochez cette case et saisissez une clause de type WHERE dans le champ **WHERE clause** fin de filtrer les données lors de l’opération d’insertion.  

**Remarque :** Il est probable que cette option ne fonctionne pas avec certaines versions des bases de données, notamment Oracle 9i. |
Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log aussi bien au niveau du Job qu’au niveau du composant. |

Global Variables

| Global Variables | NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. |
| | NB_LINE_MERGED : nombre de lignes fusionnées. Cette variable est une variable After et retourne un nombre entier. |
| | QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères. |
| | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
| | Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |
| | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
| | Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

Utilisation

| Règle d’utilisation | Ce composant est un composant intermédiaire. Vous pouvez l’utiliser avec d’autres composants de base de données, notamment les composants de connexion et de commit. |
| SQL Template | SQL Template List |

Scénario : Effectuer des opérations de MERGE directement dans le SGBD

Ce scénario décrit un Job simple permettant d’ouvrir une connexion à une base de données MySQL, d’effectuer des opérations d’insertion ou de mise à jour sur les données d’une table source vers une table cible en fonction de l’identifiant de l’utilisateur et d’afficher le contenu de la table cible avant et après les opérations de MERGE. Une clause de type WHERE est utilisée pour filtrer les données lors de ces opérations.
• À partir de la **Palette**, déposez les composants suivants dans l’espace de modélisation graphique : un **tMysqlConnection**, un **tSQLTemplateMerge**, deux **tMysqlInput** et deux **tLogRow**.

• Connectez le composant **tMysqlConnection** au premier **tMysqlInput** à l’aide d’un lien **Trigger > OnSubjobOk**.

• Connectez le premier **tMysqlInput** au premier **tLogRow** à l’aide d’un lien **Row > Main**. Cette ligne affichera le contenu initial de la table cible dans la console de la vue **Run**.

• Connectez le premier **tMysqlInput** au composant **tSQLTemplateMerge**, et le **tSQLTemplateMerge** au second composant **tMysqlInput** à l’aide de connexions **Trigger > OnSubjobOk**.

• Connectez le second **tMysqlInput** au second **tLogRow** à l’aide d’un lien **Row > Main**. Cette ligne affichera le résultat des opérations de MERGE dans la console de la vue **Run**.

• Double-cliquez sur le composant **tMysqlConnection** pour afficher l’onglet **Basic settings** de sa vue **Component**.

### tMysqlConnection_1

<table>
<thead>
<tr>
<th>Basic settings</th>
<th>Property Type</th>
<th>Built-In</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Version</td>
<td>Mysql 5</td>
<td></td>
</tr>
<tr>
<td>Host</td>
<td>localhost</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>3306</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td><em>talend</em></td>
<td></td>
</tr>
<tr>
<td>Username</td>
<td>root</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><em>talend</em></td>
<td></td>
</tr>
</tbody>
</table>

• Paramétrez manuellement les détails de la connexion à la base de données ou sélectionnez **Repository** dans la liste **Property Type** et sélectionnez votre connexion à une base de données si celle-ci a déjà été définie et stockée dans la zone **Metadata** de la vue **Repository**.
Pour plus d’informations concernant les Métadonnées, consultez le Guide utilisateur du Studio Talend.

- Double-cliquez sur le premier composant tMysqlInput pour afficher l’onglet Basic settings de sa vue Component.

  ![tMysqlInput_1](image)

  **Basic settings**

  - Cochez la case **Use an existing connection**. Si vous souhaitez utiliser plus d’un composant de connexion dans votre Job, sélectionnez le composant que vous souhaitez utiliser à partir de la liste Component List.

  - Cliquez sur le bouton [...] à côté de **Edit schema** et définissez la structure des données de la table cible, ou sélectionnez Repository à partir de la liste Schema puis sélectionnez la table cible si le schéma a déjà été défini et stocké dans la zone Metadata de la vue Repository.

Pour ce scénario, choisissez **Built-in**.

  ![Schema of tMysqlInput_1](image)

  - Définissez les colonnes comme indiqué ci-dessus, puis cliquez sur **OK** pour propager la structure du schéma au composant de sortie et fermer la boîte de dialogue.

  - Renseignez le champ **Table Name** avec le nom de la table cible, soit customer_info_merge pour ce scénario.
• Cliquez sur le bouton **Guess Query**, ou saisissez "SELECT * FROM customer_info_merge" dans la zone **Query** afin de récupérer toutes les colonnes des tables.

• Définissez les propriétés du second composant **tMysqlInput** en utilisant exactement les mêmes paramètres que dans le premier composant **tMysqlInput**.

• Dans l’onglet **Basic settings** de la vue **Component** de chaque composant **tLogRow**, sélectionnez l’option **Table** dans la zone **Mode** afin d’afficher le résultat sous forme de tableau dans la console de la vue **Run**.

• Double-cliquez sur le composant **tSQLTemplateMerge** pour afficher l’onglet **Basic settings** de sa vue **Component**.

• Saisissez les noms de la table source et de la table cible dans les champs correspondants.

Pour ce scénario, la table source, **new_customer_info**, contient huit enregistrements ; la table cible, **customer_info_merge**, contient cinq enregistrements et les deux tables ont la même structure.

**Remarque** :

Il est possible que la table source et la table cible aient des schémas différents. Dans ce cas, cependant, assurez-vous que la colonne source et la colonne cible spécifiées dans chaque ligne des tables **Merge ON**, **UPDATE Columns** et **INSERT Columns** aient un type de données identique et que la longueur de la colonne cible permette l’insertion de données à partir de la colonne source correspondante.

• Définissez le schéma source manuellement, ou sélectionnez **Repository** à partir de la liste **Schema** et sélectionnez la table souhaitée si le schéma a déjà été défini et stocké dans la zone **Metadata** de la vue **Repository**.

Pour ce scénario, des schémas en mode **Built-in** sont utilisés.
Définissez les colonnes comme indiqué ci-dessus puis cliquez sur OK pour fermer la boîte de dialogue. Faites de même pour le schéma cible.

Cliquez sur le bouton [+ ] sous la table Merge ON pour ajouter une ligne et sélectionnez la colonne ID comme clé primaire.

Cochez la case Use UPDATE (WHEN MATCHED) pour mettre à jour les données existantes lors des opérations de MERGE, définissez les colonnes à mettre à jour en cliquant sur le bouton [+] et sélectionnez les colonnes souhaitées.

Le but de ce scénario est de mettre à jour toutes les colonnes selon l’identifiant (ID) des utilisateurs. Pour cela, sélectionnez toutes les colonnes sauf la colonne ID.

Avertissement :
Les colonnes définies comme clés primaires ne peuvent pas et ne doivent pas être mises à jour.

Cochez la case Specify UPDATE WHERE clause et saisissez customer_info_merge.ID >= 4 entre guillemets doubles, dans le champ WHERE clause afin que seuls les enregistrements existants ayant un ID égal ou supérieur à 4 soient mis à jour.
Cochez la case **Use INSERT** puis définissez dans la table **INSERT Columns** les colonnes où les données seront prises et insérées.

Pour ce scénario, tous les enregistrements qui n'existent pas dans la table cible sont insérés.

- Sélectionnez l'onglet **SQL Template** pour afficher et ajouter les modèles SQL à utiliser.

Par défaut, le composant **SQLTemplateMerge** utilise deux modèles SQL système : **MergeUpdate** et **MergeInsert**.

**Remarque :**

Dans l'onglet **SQL Template**, vous pouvez ajouter des modèles SQL système ou créer vos propres modèles et les utiliser dans votre Job pour effectuer des opérations codées. Pour plus d'informations, consultez **Propriétés du tSQLTemplateFilterColumns Standard** à la page 3890.
• Cliquez sur le bouton **Add** pour ajouter une ligne puis sélectionnez **Commit** dans la liste des modèles afin de committer les résultats des opérations de MERGE dans votre base de données.

Autrement, vous pouvez connecter le composant **tSQLTemplateMerge** à un **tSQLTemplateCommit** ou un **tMySqlCommit** à l'aide d'un lien **Trigger > OnSubjobOk** afin de commiter les résultats des opérations dans votre base de données.

• Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

La console de la vue **Run** affiche à la fois le contenu d'origine de la table cible ainsi que les résultats des opérations de MERGE. Dans la table cible, les enregistrements No. 4 et No. 5 contiennent les informations mises à jour, tandis que les enregistrements No. 6, No. 7 et No. 8 contiennent les informations insérées dans la table.

```sql
Starting job MergeData at 10:56 01/04/2011.
[statistics] connecting to socket on port 3580
[statistics] connected

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Birth Date</th>
<th>Address</th>
<th>City</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Calvin Wilson</td>
<td>06-04-2008</td>
<td>997 East Calle Prima</td>
<td>Topeka</td>
<td>93007</td>
</tr>
<tr>
<td>2</td>
<td>Benjamin Johnson</td>
<td>12-01-2007</td>
<td>366 Newbury Road</td>
<td>Olympia</td>
<td>94101</td>
</tr>
<tr>
<td>3</td>
<td>Millard Hayes</td>
<td>03-03-2007</td>
<td>411 San Marcos</td>
<td>Carson City</td>
<td>90207</td>
</tr>
<tr>
<td>4</td>
<td>George Quincy</td>
<td>01-05-2000</td>
<td>516 Carpenteria South</td>
<td>Bismarck</td>
<td>5807</td>
</tr>
<tr>
<td>5</td>
<td>Ronald Coolidge</td>
<td>12-01-2008</td>
<td>780 Newbury Road</td>
<td>Jackson</td>
<td>90215</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Birth Date</th>
<th>Address</th>
<th>City</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Calvin Wilson</td>
<td>06-04-2008</td>
<td>997 East Calle Prima</td>
<td>Topeka</td>
<td>93007</td>
</tr>
<tr>
<td>2</td>
<td>Benjamin Johnson</td>
<td>12-01-2007</td>
<td>366 Newbury Road</td>
<td>Olympia</td>
<td>94101</td>
</tr>
<tr>
<td>3</td>
<td>Millard Hayes</td>
<td>03-03-2007</td>
<td>411 San Marcos</td>
<td>Carson City</td>
<td>90207</td>
</tr>
<tr>
<td>4</td>
<td>George Quincy</td>
<td>01-05-2000</td>
<td>516 Carpenteria South</td>
<td>Bismarck</td>
<td>5807</td>
</tr>
<tr>
<td>5</td>
<td>Ronald Coolidge</td>
<td>12-01-2008</td>
<td>780 Newbury Road</td>
<td>Jackson</td>
<td>90215</td>
</tr>
</tbody>
</table>

[statistics] disconnected
Job MergeData ended at 10:56 01/04/2011. [exit code=0]
tSQLTemplateRollback

Ce composant annule la transaction commitée dans la base de données SQLTemplate connectée.

Propriétés du tSQLTemplateRollback Standard

Ces propriétés sont utilisées pour configurer le tSQLTemplateRollback s'exécutant dans le framework de Jobs Standard.

Le composant tSQLTemplateRollback Standard appartient à la famille ELT.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Database Type</th>
<th>Sélectionnez dans le menu déroulant le type de base de données sur lequel vous voulez travailler.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component List</td>
<td>Sélectionnez le composant ELT de connexion le plus pertinent dans la liste si vous prévoyez d'ajouter plus d'une connexion à votre Job en cours.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décrochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Global Variables

| Global Variables | NB_LINE : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier. QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères. ERROR_MESSAGE : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l'exécution d'un composant. Une variable After fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |

3905
Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en association avec des composants ELT, notamment avec le <em>tSQLTemplateCommit</em> et le composant ELT de connexion correspondant à la base de données avec laquelle vous travaillez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Template</td>
<td>SQLTemplate List</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario associé au *tSQLTemplateRollback*, consultez Scénario : Filtre et agréger les colonnes d’une table directement dans le SGBD à la page 3883.
**tSqoopExport**

Ce composant définit les arguments requis par Sqoop pour transférer des données dans un SGBDR.

Le tSqoopExport appelle Sqoop pour transférer des données du système de fichiers distribué Hadoop (Hadoop Distributed File System, HDFS) vers un système de gestion de base de données relationnelle (SGBDR).

Certaines fonctionnalités fournies par ce composant sont supportées uniquement par la dernière version de Sqoop. Pour plus d’informations concernant la disponibilité de chaque fonctionnalité, consultez la documentation Apache à propos de Sqoop.

**Remarque :**

Sqoop est installé dans chaque distribution Hadoop. Cependant, si la distribution Hadoop que vous utilisez n’a pas de Sqoop installé, vous devez en installer un et vous assurer d’ajouter l’invite de commande Sqoop dans la variable PATH de la distribution. Pour plus d’informations concernant l’installation de Sqoop, consultez la documentation de Sqoop.

**Arguments supplémentaires**

<table>
<thead>
<tr>
<th>Mode Commandline</th>
<th>Mode Java API</th>
</tr>
</thead>
<tbody>
<tr>
<td>--driver</td>
<td>jdbc.driver.class</td>
</tr>
<tr>
<td>--direct-split-size</td>
<td>import.direct.split.size</td>
</tr>
<tr>
<td>--inline-lob-limit</td>
<td>import.max.inline.lob.size</td>
</tr>
<tr>
<td>--split-by</td>
<td>db.split.column</td>
</tr>
<tr>
<td>--warehouse-dir</td>
<td>hdfs.warehouse.dir</td>
</tr>
<tr>
<td>--enclosed-by</td>
<td>codegen.output.delimiters.enclose</td>
</tr>
<tr>
<td>--escaped-by</td>
<td>codegen.output.delimiters.escape</td>
</tr>
<tr>
<td>--fields-terminated-by</td>
<td>codegen.output.delimiters.field</td>
</tr>
<tr>
<td>--lines-terminated-by</td>
<td>codegen.output.delimiters.record</td>
</tr>
<tr>
<td>--optionally-enclosed-by</td>
<td>codegen.output.delimiters.required</td>
</tr>
<tr>
<td>--input-enclosed-by</td>
<td>codegen.input.delimiters.enclose</td>
</tr>
<tr>
<td>--input-escaped-by</td>
<td>codegen.input.delimiters.escape</td>
</tr>
<tr>
<td>--input-fields-terminated-by</td>
<td>codegen.input.delimiters.field</td>
</tr>
<tr>
<td>--input-lines-terminated-by</td>
<td>codegen.input.delimiters.record</td>
</tr>
<tr>
<td>--input-optionally-enclosed-by</td>
<td>codegen.input.delimiters.required</td>
</tr>
<tr>
<td>--hive-home</td>
<td>hive.home</td>
</tr>
</tbody>
</table>
**Propriétés du tSqoopExport Standard**

Ces propriétés sont utilisées pour configurer le tSqoopExport s'exécutant dans le framework de Jobs Standard.

Le composant tSqoopExport Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l'une des solutions Big Data de Talend.

**Basic settings**

<table>
<thead>
<tr>
<th>Mode Commandline</th>
<th>Mode Java API</th>
</tr>
</thead>
<tbody>
<tr>
<td>--hive-import</td>
<td>hive.import</td>
</tr>
<tr>
<td>--hive-overwrite</td>
<td>hive.overwrite.table</td>
</tr>
<tr>
<td>--hive-table</td>
<td>hive.table.name</td>
</tr>
<tr>
<td>--class-name</td>
<td>codegen.java.classname</td>
</tr>
<tr>
<td>--jar-file</td>
<td>codegen.jar.file</td>
</tr>
<tr>
<td>--outdir</td>
<td>codegen.output.dir</td>
</tr>
<tr>
<td>--package-name</td>
<td>codegen.java.packagename</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant les arguments disponibles pour le mode Commandline de Sqoop, consultez la documentation de Sqoop.

exemple, les versions sont les mêmes ou la version JDK de la machine Hadoop est plus récente.

| Hadoop properties | Peut être **Built-in** ou **Repository** :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>Built-in</strong> : saisissez les informations de configuration de la distribution Hadoop à utiliser localement, pour ce composant uniquement.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : vous avez déjà créé la connexion Hadoop et l’avez stockée dans le <strong>Repository</strong> ; vous pouvez la réutiliser directement pour la configuration du composant et la création d’un Job. Pour plus d’informations concernant la création d’une connexion centralisée Hadoop, consultez <strong>Guide de prise en main de Talend Open Studio for Big Data</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Si disponible dans la liste de <strong>Distribution</strong>, l’option <strong>Microsoft HD Insight</strong> vous permet d’utiliser un cluster Microsoft HD Insight. Dans cette optique, vous devez configurer les connexions au cluster HD Insight et au service Windows Azure Storage du cluster dans les zones affichées. Pour plus d’informations concernant ces paramètres, recherchez Configurer manuellement la connexion, sur Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>)..&lt;br&gt;• Si vous sélectionnez <strong>Amazon EMR</strong>, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).&lt;br&gt;• L’option <strong>Custom</strong> vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par <strong>Talend</strong>.</td>
</tr>
</tbody>
</table>

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,<br>2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.<br>Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip
Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop Version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameNode URI</td>
<td>Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée masternode comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.</td>
</tr>
<tr>
<td>JobTracker Host</td>
<td>Cochez cette case et, dans le champ qui s’affiche, saisissez l’emplacement du ResourceManager de votre distribution. Par exemple tal-qall4.talend.lan:8050. Cette propriété est requise lorsque la requête que vous souhaitez utiliser est exécutée dans Windows et est une requête Select. Par exemple, SELECT your_column_name FROM your_table_name Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d’un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du de l’exécution) :</td>
</tr>
</tbody>
</table>
1. Cochez la case **Set resourcemanager scheduler address** et saisissez l’adresse de l’ordonnanceur (Scheduler) dans le champ qui apparaît.

2. Cochez la case **Set jobhistory address** et saisissez l’emplacement du serveur JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les informations relatives aux métriques du Job courant sur le serveur JobHistory.

3. Cochez la case **Set staging directory** et saisissez le chemin d’accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l’exécution de programmes. Ce répertoire se trouve sous la propriété `yarn.app.mapreduce.am.staging-dir` dans les fichiers de configuration, notamment les fichiers `yarn-site.xml` et `mapred-site.xml` de votre distribution.

4. Allouez des volumes de mémoire aux calculs **Map** et **Reduce** et au service **ApplicationMaster** de YARN en cochant la case **Set memory** dans la vue **Advanced settings**.

5. Cochez la case **Set Hadoop user** et saisissez le nom de l’utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d’écriture, ce champ vous permet d’exécuter le Job directement avec l’utilisateur ayant les droits d’accès appropriés au fichier ou répertoire à traiter.

6. Cochez la case **Use datanode hostname** pour permettre au Job d’accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété `dfs.client.use.datanode.hostname` à `true`. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.


---

**Use kerberos authentication**

Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le "principal name" de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans [Connexion sécurisée à MapR](#) à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées
les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

De plus, comme ce composant effectue des calculs Map/Reduce, vous devez également authentifier les services associés, comme le serveur de l'historique des Jobs et le gestionnaire de ressources ou le JobTracker, selon votre distribution, dans le champ correspondant. Ces principaux se trouvent dans les fichiers de configuration de votre distribution. Par exemple, dans une distribution CDH4, le principal du gestionnaire de ressource est configuré dans le fichier `yarn-site.xml` et le principal de l'historique des Job dans le fichier `mapred-site.xml`.

Cette case est disponible ou indisponible selon la distribution d'Hadoop à laquelle vous vous connectez.

### Use a keytab to authenticate

Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l’aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est `user1` et le principal à utiliser est `guest`. Dans cette situation, assurez-vous que `user1` a les droits de lecture pour le fichier keytab à utiliser.

### Hadoop user name

Saisissez le nom de l’utilisateur avec lequel vous souhaitez exécuter le Job. Puisqu’un fichier ou un répertoire dans Hadoop a son auteur spécifique, avec les droits en lecture ou écriture appropriés, ce champ vous permet d’exécuter le Job directement sous le nom d’utilisateur ayant les droits appropriés pour accéder au fichier ou au répertoire à traiter. Notez que ce champ peut n’être pas disponible selon la distribution que vous utilisez.

### JDBC property

Peut être **Built-in** ou **Repository**:
- **Built-in** : saisissez les informations de connexion à la base de données à utiliser localement, pour ce composant uniquement.
- **Repository** : vous devez avoir créé la connexion à la base de données et l’avoir stockée dans le Repository; vous pouvez la réutiliser directement pour la configuration du composant et la création d’un Job. Pour plus d’informations concernant la création d’une connexion à la base de données centralisée, consultez le [Guide utilisateur du Studio Talend](#).
Notez que seules les connexions de type General JDBC stockées dans le Repository sont supportées.

<table>
<thead>
<tr>
<th><strong>Connection</strong></th>
<th>Saisissez l’URL JDBC utilisée pour vous connecter à la base de données cible.</th>
</tr>
</thead>
</table>
| **User name et Password** | Saisissez les informations d’authentification à la base de données cible.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.  
Si votre mot de passe est stocké dans un fichier, cochez la case The password is stored in a file et saisissez le chemin d’accès à ce fichier dans le champ File path qui s’affiche.  
• Ce fichier peut être stocké sur la machine où le Job est exécuté ou dans le système HDFS du cluster Hadoop à utiliser.  
• Le mot de passe stocké dans ce fichier ne doit pas contenir ‘\n’ (nouvelle ligne) à la fin. Vous ne devez pas insérer de nouvelle ligne, même vide, à la fin du mot de passe. |
| **Driver JAR** | Quel que soit le mode, Use Commandline ou Java API, vous devez ajouter le fichier du pilote de la base de données à utiliser, dans le dossier lib de la distribution Hadoop que vous utilisez. Pour ce faire, utilisez la table Driver JAR afin d’ajouter ce fichier de pilote pour le Job en cours de construction. |
| **Table Name** | Saisissez le nom de la table cible dans laquelle les données sont transférées depuis HDFS. Cette table doit déjà exister dans la base de données cible. Les fichiers d’entrée sont lus et parsez en un ensemble d’enregistrements selon les séparateurs personnalisés. |
| **Export Dir** | Saisissez le chemin d’accès ou parcourez votre répertoire jusqu’aux données sources devant être transférées dans HDFS. |
| **Direct** | Cochez cette case pour utiliser le chemin d’export rapide. |
| **Specify Number of Mappers** | Cochez cette case afin de préciser le nombre de tâches de "map" (processus parallèles) utilisé pour effectuer le transfert des données.  
Si ne vous souhaitez pas que Sqoop travaille en parallèle, saisissez 1 dans le champ affiché. |
| **Call a stored procedure** | Cochez cette case pour permettre au composant d’appeler une procédure stockée spécifique pour écrire des données dans la base de données cible. |
Vous devez saisir le nom de la procédure stockée à utiliser dans le champ affiché.

La syntaxe et les fonctionnalités des procédures stockées varient selon les bases de données. Il est donc recommandé de consultez la documentation de la base de données que vous souhaitez utiliser, pour plus d’informations au sujet de la procédure stockée que vous souhaitez appeler.

**Use batch mode**

Cochez cette case pour exécuter les instructions par lots et non en exécutant une instruction d’insertion (INSERT) multiligne pour écrire différents enregistrements dans une base de données cible.

**Clear staging table**

Si vous utilisez une table de préparation spécifique pour le transfert de données souhaité, cochez cette case pour vous assurer que la table de préparation est vide lorsque le transfert de données s’exécute.

**Define a staging table**

Cochez cette case pour créer une table de préparation pour les données à transférer. Les données sont transférées dans cette table avant d’être écrites dans la table cible, afin d’éviter que seule une partie des données soit committée dans la table cible si le transfert échoue.

Pour plus d’informations concernant le support d’une table de préparation pour un transfert de données, consultez la documentation Apache pour Sqoop.

**Specify how updates are performed when new rows are found with non-match keys in database**

Cochez cette case pour déterminer l’action à effectuer lorsqu’une clé de mise à jour ne correspond à aucun enregistrement dans la table cible. Vous pouvez sélectionner l’une des options suivantes :

- **Update only** : met à jour uniquement les enregistrements existant déjà dans la table cible.
- **Allow insert** : fonctionne comme l'instruction SQL UPSERT. Cette option permet d’écrire de nouveaux enregistrements dans la table s’ils n’existent pas.

**Use column for update**

Cochez cette case et, dans la table qui s’affiche, ajoutez les colonnes à utiliser comme clés de mise à jour.

**Print Log**

Cochez cette case pour activer la case **Verbose**.

**Verbose**

Cochez cette case pour imprimer plus d’informations pendant que vous travaillez, par exemple des informations de débogage.

### Advanced settings

**Use MySQL default delimiters**

Cochez cette case pour utiliser les séparateurs MySQL par défaut. Cette case est disponible en mode Commandline (**Use Commandline**).

**Define Java mapping**

Sqoop fournit une configuration par défaut, mappant la plupart des types SQL aux types Hive correspondants.
Si vous devez utiliser votre mapping personnalisé, pour écraser les mappings par défaut au moment de l'exécution, cochez cette case et définissez les mappings à utiliser dans la table qui apparaît.

### Additional arguments

Complétez ce tableau pour utiliser des arguments supplémentaires, si nécessaire.

En ajoutant des arguments supplémentaires, vous pouvez effectuer de multiples opérations en une seule transaction. Par exemple, vous pouvez utiliser `--hive-import` et `--hive-table` en mode Commandline *(Use Commandline)* ou `hive.import` et `hive.table.name` en mode Java API *(Use Java API)* pour créer une table Hive et écrire des données dans cette table lors de l'exécution de la transaction écrivant des données dans HDFS. Pour plus d’informations concernant les arguments Sqoop disponibles en mode Commandline et en mode Java API, consultez Arguments supplémentaires à la page 3907.

### Use speed parallel data transfers

Cochez cette case pour permettre des transferts de données rapides et parallèles entre la base de données Teradata et la distribution Hadoop Hortonworks.

La table **Specific params** et la case **Use additional params** apparaissent vous permettant de spécifier les paramètres requis pour les transferts parallèles.

- Dans la table **Specific params**, deux colonnes sont disponibles :
  - **Argument** : dans la liste, sélectionnez les paramètres selon vos besoins. Ce sont les paramètres les plus communs pour les transferts parallèles.
  - **Value** : saisissez la valeur des paramètres.

- En cochant la case **Additional params**, le champ **Specific additional params** s’affiche. Dans ce champ, vous pouvez saisir les paramètres Teradata que vous souhaitez utiliser mais qui ne sont pas modifiables dans la table **Specific params**. La syntaxe de ce paramètre est `-Dparameter=value`. Si vous saisissez plus d’un paramètre dans ce champ, séparez-les à l’aide d’un espace.

Vous devez vous assurer que Hortonworks Connector for Teradata a été installé dans votre cluster Hortonworks.


Cette option est disponible uniquement un mode **Use Commandline**.

### Hadoop properties

Le **Studio Talend** utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser
une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

• Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :


| **Mapred job map memory mb** et **Mapred job reduce memory mb** | Vous pouvez personnaliser les opérations map et reduce en cochant la case **Set memory**, pour configurer les allocations de mémoire pour ces opérations à effectuer par le système Hadoop.

Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations **map** et **reduce** dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux **1000**, ce qui est normalement adapté pour l’exécution de ces opérations.

Les paramètres de mémoire à définir sont **Map (in Mb)**, **Reduce (in Mb)** et **ApplicationMaster (in Mb)**. Ces champs permettent d’allouer dynamiquement de la mémoire aux opérations map et reduce et à l’ApplicationMaster de YARN. |
| **Path separator in server** | Laissez le champ **Path separator in server** tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d’autres termes, changez le séparateur si celui-ci n’est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau des composants. |
Variables globales


Utilisation


| Prérequis | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR. • Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les bibliothèques du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). Si vous n’ajoutez pas de bibliothèque, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path. |
Configurez l’argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue **[Preferences]** dans le menu **Window**. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (**Data viewer**) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

**Limitation**

Si vous avez sélectionné le mode **Use Commandline**, vous devez utiliser l’hôte où Sqoop est installé pour exécuter le Job à l’aide de ce composant.

**Connexions**

Liens de sortie (de ce composant à un autre) :
- **Trigger** : Run if, On Subjob Ok, On Subjob Error.

Liens d’entrée (d’un autre composant à celui-ci) :
- **Row** : Iterate,

Pour plus d’informations concernant les connexions, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
Ce composant définit les arguments requis par Sqoop pour écrire les données qui vous intéressent dans HDFS.

Le tSqoopImport appelle Sqoop pour transférer des données d’un système de gestion de base de données relationnelle (SGBDR) comme MySQL ou Oracle dans le système de fichiers distribué Hadoop (Hadoop Distributed File System, HDFS).

**Remarque :**

Sqoop est installé dans chaque distribution Hadoop. Cependant, si la distribution Hadoop que vous utilisez n’a pas de Sqoop installé, vous devez en installer un et vous assurer d’ajouter l’invite de commande Sqoop dans la variable PATH de la distribution. Pour plus d’informations concernant l’installation de Sqoop, consultez la documentation de Sqoop.

### Propriétés du tSqoopImport Standard

Ces propriétés sont utilisées pour configurer le tSqoopImport s’exécutant dans le framework de Jobs Standard.

Le composant tSqoopImport Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

#### Basic settings

<table>
<thead>
<tr>
<th>Mode</th>
<th>Sélectionnez le mode dans lequel Sqoop est appelé dans l’exécution du Job.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Commandline</strong></td>
<td>l’interpréteur de commandes Sqoop est utilisé pour appeler Sqoop. Dans ce mode, vous devez déployer et exécuter le Job dans l’hôte où Sqoop est installé. Si votre solution Talend est soumise à souscription, il est recommandé d’installer et d’utiliser un JobServer fourni par Talend dans cet hôte pour exécuter le Job. Si vous utilisez Talend Open Studio for Big Data, assurez-vous que le Studio Talend et Sqoop se trouvent sur la même machine.</td>
</tr>
<tr>
<td><strong>Use Java API</strong></td>
<td>l’API Java est utilisée pour appeler Sqoop. Dans ce mode, le Job peut être exécuté localement dans le Studio Talend mais vous devez configurer la connexion à la distribution Hadoop à utiliser. Notez que JDK est requis pour l’exécution du Job en mode API Java et que les versions des kits JDK installés sur les deux machines doivent être compatibles entre elles. Par exemple, les versions sont les mêmes ou la version JDK de la machine Hadoop est plus récente.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hadoop properties</th>
<th>Peut être Built-in ou Repository :</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Built-in</td>
<td>saisissez les informations de configuration de la distribution Hadoop à utiliser localement, pour ce composant uniquement.</td>
</tr>
</tbody>
</table>
**Repository** : vous avez déjà créé la connexion Hadoop et l'avez stockée dans le Repository ; vous pouvez la réutiliser directement pour la configuration du composant et la création d'un Job. Pour plus d'informations concernant la création d'une connexion centralisée Hadoop, consultez *Guide de prise en main de Talend Open Studio for Big Data*.

### Distribution

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.

- Si disponible dans la liste de Distribution, l'option **Microsoft HD Insight** vous permet d'utiliser un cluster Microsoft HD Insight. Dans cette optique, vous devez configurez les connexions au cluster HD Insight et au service Windows Azure Storage du cluster dans les zones affichées. Pour plus d’informations concernant ces paramètres, recherchez Configurer manuellement la connexion, sur Talend Help Center (*https://help.talend.com*).

- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (*https://help.talend.com*).

- L’option **Custom** vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l'évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d'utiliser l'option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend**, **Talend** et sa Communauté fournissent l'opportunité de vous
connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop Version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>NameNode URI</th>
<th>Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée masternode comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.</th>
</tr>
</thead>
</table>

| JobTracker Host | Cochez cette case et, dans le champ qui s’affiche, saisissez l’emplacement du ResourceManager de votre distribution. Par exemple tal-qall4.talend.lan:8050.  
Cette propriété est requise lorsque la requête que vous souhaitiez utiliser est exécutée dans Windows et est une requête Select. Par exemple, SELECT your_column_name FROM your_table_name  
Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d’un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du de l’exécution) :  
1. Cochez la case Set resourcemanager scheduler address et saisissez l’adresse de l’ordonnanceur (Scheduler) dans le champ qui apparaît.  
3. Cochez la case **Set staging directory** et saisissez le chemin d'accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l'exécution de programmes. Ce répertoire se trouve sous la propriété `yarn.app.mapreduce.am.staging-dir` dans les fichiers de configuration, notamment les fichiers `yarn-site.xml` et `mapred-site.xml` de votre distribution.

4. Allouez des volumes de mémoire aux calculs **Map** et **Reduce** et au service **ApplicationMaster** de **YARN** en cochant la case **Set memory** dans la vue **Advanced settings**.

5. Cochez la case **Set Hadoop user** et saisissez le nom de l'utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d'écriture, ce champ permet d'exécuter le Job directement avec l’utilisateur ayant les droits d'accès appropriés au fichier ou répertoire à traiter.

6. Cochez la case **Use datanode hostname** pour permettre au Job d'accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété `dfs.client.use.datanode.hostname` à true. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.


---

### Use kerberos authentication

Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le 'principal name' de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans [Connexion sécurisée à MapR](#) à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

De plus, comme ce composant effectue des calculs Map/Reduce, vous devez également authentifier les services associés, comme le serveur de l’historique des Jobs et le gestionnaire de ressources ou le JobTracker, selon votre distribution, dans le champ correspondant. Ces principaux se trouvent dans les fichiers de configuration...
de votre distribution. Par exemple, dans une distribution CDH4, le principal du gestionnaire de ressource est configuré dans le fichier `yarn-site.xml` et le principal de l'historique des Job dans le fichier `mapred-site.xml`. Cette case est disponible ou indisponible selon la distribution d'Hadoop à laquelle vous vous connectez.

| **Use a keytab to authenticate** | Cochez la case **Use a keytab to authenticate** pour vous connecter à un système utilisant Kerberos à l'aide d'un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d'accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend. Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est **user1** et le principal à utiliser est **guest**. Dans cette situation, assurez-vous que **user1** a les droits de lecture pour le fichier keytab à utiliser. |
| **Hadoop user name** | Saisissez le nom de l'utilisateur avec lequel vous souhaitez exécuter le Job. Puisqu'un fichier ou un répertoire dans Hadoop a son auteur spécifique, avec les droits en lecture ou écriture appropriés, ce champ vous permet d'exécuter le Job directement sous le nom d'utilisateur ayant les droits appropriés pour accéder au fichier ou au répertoire à traiter. Notez que ce champ peut n'être pas disponible selon la distribution que vous utilisez. |
| **JDBC property** | Peut être **Built-in** ou **Repository** :
• **Built-in** : saisissez les informations de connexion à la base de données à utiliser localement, pour ce composant uniquement.
• **Repository** : vous devez avoir créé la connexion à la base de données et l'avoir stockée dans le Repository, vous pouvez la réutiliser directement pour la configuration du composant et la création d'un Job. Pour plus d'informations concernant la création d'une connexion à la base de données centralisée, consultez le Guide utilisateur du Studio Talend . Notez que seules les connexions de type **General JDBC** stockées dans le **Repository** sont supportées. |
| **Connection** | Saisissez l'URL JDBC utilisée pour vous connecter à la base de données dans laquelle les données source sont stockées. |
| **User name et Password** | Saisissez les informations d'authentification utilisées pour vous connecter à la base de données source. |
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

Si votre mot de passe est stocké dans un fichier, cochez la case The password is stored in a file et saisissez le chemin d'accès à ce fichier dans le champ File path qui s’affiche.

- Ce fichier peut être stocké sur la machine où le Job est exécuté ou dans le système HDFS du cluster Hadoop à utiliser.
- Le mot de passe stocké dans ce fichier ne doit pas contenir "\n" (nouvelle ligne) à la fin. Vous ne devez pas insérer de nouvelle ligne, même vide, à la fin du mot de passe.

<table>
<thead>
<tr>
<th>Driver JAR</th>
<th>Quel que soit le mode, Use Commandline ou Java API, vous devez ajouter le fichier du pilote de la base de données à utiliser, dans le dossier lib de la distribution Hadoop que vous utilisez. Pour ce faire, utilisez la table Driver JAR afin d’ajouter ce fichier de pilote pour le Job en cours de construction.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name</td>
<td>Saisissez le nom de la table à transférer dans HDFS. Ce champ est indisponible lorsque vous utilisez le mode de requêtes de forme libre, en cochant la case Use query.</td>
</tr>
</tbody>
</table>
| File format | Sélectionnez un format de fichier pour les données à transférer:  
• textfile  
• sequencefile  
• Avro file  
• Parquet file: la version de Sqoop doit être 1.4.6. |
| Delete target directory | Cochez cette case pour supprimer le répertoire cible du transfert. |
| Append | Cochez cette case pour écrire à la suite d’un ensemble de données existant dans HDFS les données transférées. |
| Compress | Cochez cette case pour activer la compression. |
| Direct | Cochez cette case pour utiliser le chemin d’import rapide. |
| Specify columns | Cochez cette case pour afficher le tableau Columns dans lequel vous souhaitez spécifier les colonnes de données que vous souhaitez transférer dans HDFS. |
| Use HERE clause | Cochez cette case pour utiliser une clause WHERE contrôlant les lignes à transférer. Dans le champ activé, saisissez la condition utilisée pour sélectionner les lignes de données. Par exemple, saisissez id > 400. |
| Use query | Cochez cette case pour utiliser le mode de requête de forme libre, fourni par Sqoop.
Une fois la case cochée, vous pouvez saisir librement la requête que vous souhaitez utiliser.
Spécifiez le répertoire cible. Si Sqoop importe les données en parallèle, spécifiez également l'argument **Split by**.

**Avertissement :**
Une fois les requêtes saisies, la valeur de l'argument **--fields-terminated-by** peut uniquement être paramétrée à "\t" dans la table **Additional arguments** de l'onglet **Advanced settings**.
 |
| Specify Target Dir | Cochez cette case afin de saisir le chemin d'accès au fichier cible dans lequel vous souhaitez transférer les données source.
Cet emplacement peut être un nouveau répertoire. Si ce n'est pas le cas, cochez la case **Append**.
 |
| Specify Split by | Cochez cette case puis saisissez le nom de la colonne que vous souhaitez utiliser comme colonne de division, afin de diviser la charge de travail.
Par exemple, pour une table où la colonne *id* est la colonne clé, saisissez *tablename.id*. Sqoop divise les données à transférer selon la valeur de leur ID et les importe en parallèle.
 |
| Specify Number of Mappers | Cochez cette case afin de préciser le nombre de tâches de "map" (processus parallèles) utilisé pour effectuer le transfert des données.
Si ne vous souhaitez pas que Sqoop travaille en parallèle, saisissez 1 dans le champ affiché.
 |
| Print Log | Cochez cette case pour activer la case **Verbose**.
 |
| Verbose | Cochez cette case pour imprimer plus d'informations pendant que vous travaillez, par exemple des informations de débogage.
 |
### Advanced settings

| Use MySQL default delimiters | Cochez cette case pour utiliser les séparateurs MySQL par défaut. Cette case est disponible en mode Commandline (**Use Commandline**).
 |
| Define Java mapping | Sqoop fournit une configuration par défaut, mappant la plupart des types SQL aux types Hive correspondants. Si vous devez utiliser votre mapping personnalisé, pour écraser les mappings par défaut au moment de }
l'exécution, cochez cette case et définissez les mappings à utiliser dans la table qui apparaît.

<table>
<thead>
<tr>
<th>Define Hive mapping</th>
<th>Cette fonctionnalité est disponible selon la version de Sqoop que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional arguments</td>
<td>Complétez ce tableau pour utiliser des arguments supplémentaires, si nécessaire. En ajoutant des arguments supplémentaires, vous pouvez effectuer de multiples opérations en une seule transaction. Par exemple, vous pouvez utiliser (--\text{h Hive-import} ) et (--\text{h Hive-table}) en mode Commandline (Use Commandline) ou (\text{h Hive.import} ) et (\text{h Hive.table.name}) en mode Java API (Use Java API) pour créer une table Hive et écrire des données dans cette table lors de l'exécution de la transaction écrivant des données dans HDFS. Pour plus d'informations concernant les arguments Sqoop disponibles en mode Commandline et en mode Java API, consultez Arguments supplémentaires à la page 3907.</td>
</tr>
</tbody>
</table>
| Use speed parallel data transfers | Cochez cette case pour permettre des transferts de données rapides et parallèles entre la base de données Teradata et la distribution Hadoop Hortonworks. La table Specific params et la case Use additional params apparaissent vous permettant de spécifier les paramètres requis pour les transferts parallèles.  
- Dans la table Specific params, deux colonnes sont disponibles :  
  - **Argument** : dans la liste, sélectionnez les paramètres selon vos besoins. Ce sont les paramètres les plus communs pour les transferts parallèles.  
  - **Value** : saisissez la valeur des paramètres.  
- En cochant la case Additional params, le champ Specific additional params s'affiche. Dans ce champ, vous pouvez saisir les paramètres Teradata que vous souhaitez utiliser mais qui ne sont pas modifiables dans la table Specific params. La syntaxe de ce paramètre est \(-\text{Dparameter=value}\). Si vous saisissez plus d'un paramètre dans ce champ, séparez-les à l'aide d'un espace.  
Cette option est disponible uniquement un mode Use Commandline. |
| Hadoop properties | Le Studio Talend utilise une configuration par défaut pour son moteur, afin d'effectuer des opérations |
Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- Notez que, si vous utilisez les métadonnées stockées centralement dans le **Repository**, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste **Property type**, vous passiez de **Repository** à **Built-in**.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :


| Mapred job map memory mb et Mapred job reduce memory mb | Vous pouvez personnaliser les opérations map et reduce en cochant la case **Set memory**, pour configurer les allocations de mémoire pour ces opérations à effectuer par le système Hadoop.  

Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations *map* et *reduce* dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux `1000`, ce qui est normalement adapté pour l’exécution de ces opérations.  

Les paramètres de mémoire à définir sont **Map (in Mb)**, **Reduce (in Mb)** et **ApplicationMaster (in Mb)**. Ces champs permettent d’allouer dynamiquement de la mémoire aux opérations map et reduce et à l’ApplicationMaster de YARN. |
| Path separator in server | Laissez le champ **Path separator in server** tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d’autres termes, changez le séparateur si celui-ci n’est pas le signe deux points (`:`). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte. |
tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau des composants.

Variables globales


Utilisation


| Prérequis | La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR. • Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais). |
Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument `-Djava.library.path`, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

Limitation
Si vous avez sélectionné le mode Use Commandline, vous devez utiliser l’hôte où Sqoop est installé pour exécuter le Job à l’aide de ce composant.

Connexions
Liens de sortie (de ce composant à un autre) :
Trigger : Run if, On Subjob Ok, On Subjob Error.

Liens d’entrée (d’un autre composant à celui-ci) :
Row : Iterate.

Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.

Scénario : Importer une table MySQL dans HDFS

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario montre comment utiliser le tSqoopImport pour importer une table MySQL dans un système HDFS donné.

Les données d’exemple utilisées dans ce scénario se présentent comme suit :

<table>
<thead>
<tr>
<th>id</th>
<th>wage</th>
<th>mod_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2000</td>
<td>2008-06-26 04:25:59</td>
</tr>
<tr>
<td>1</td>
<td>2300</td>
<td>2011-06-12 05:29:45</td>
</tr>
<tr>
<td>3</td>
<td>3000</td>
<td>2010-05-02 15:34:05</td>
</tr>
</tbody>
</table>
Les données sont stockées dans une table MySQL nommée sqoopmerge.

Avant de commencer à reproduire ce scénario, vérifiez que vous disposez des droits et permissions nécessaires pour accéder à la distribution Hadoop utilisée. Ensuite, procédez comme suit :

**Déposer le composant**

**Procédure**

1. Dans la perspective Integration du Studio, créez un Job vierge depuis le nœud Job Designs du Repository.
   
   Pour plus d’informations sur la création d’un Job, consultez le Guide utilisateur du Studio Talend.

2. Déposer un **tSqoopImport** dans l’espace de modélisation graphique.

**Importer la table MySQL**

Pourquoi et quand exécuter cette tâche

Configurer le tSqoopImport

**Procédure**

1. Double-cliquez sur le **tSqoopImport** afin d’ouvrir sa vue Component.
2. Dans la zone **Mode**, sélectionnez **Use Java API**.

3. Dans la zone **Version**, sélectionnez la distribution Hadoop à utiliser et sa version. Si vous ne trouvez pas la distribution correspondant à la vôtre dans la liste, sélectionnez **Custom** afin de vous connecter à une distribution Hadoop non officiellement supportée par le Studio.

   Pour un exemple d’utilisation étape par étape de cette option **Custom**, consultez **Connexion à une distribution Hadoop personnalisée** à la page 1677.

4. Dans le champ **NameNode URI**, saisissez l’emplacement du nœud maître, le NameNode, de la distribution utilisée. Si vous utilisez WebHDFS, l’emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être `swebhdfs` et vous devez
utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.


   Notez que le mot "Job" dans le terme JobTracker désigne les Jobs MR ou Map/Reduce décrits dans la documentation d’Apache disponible sur le site Internet d’Apache http://hadoop.apache.org/ (en anglais).


   Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case Use a keytab to authenticate. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

   Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser.


8. Dans les champs Username et Password, saisissez les informations d’authentification.

9. Sous la table Driver JAR, cliquez sur le bouton [+] pour ajouter une ligne. Dans cette ligne, cliquez sur le bouton [...] pour afficher la liste déroulante et sélectionnez le fichier Jar à utiliser. Dans ce scénario, sélectionnez mysql-connector-java-5.1.30-bin.jar.

   Si le bouton [...] n’est pas disponible, cliquez dans la ligne pour le faire apparaître.

10. Dans le champ Table Name, saisissez le nom de la table source. Dans ce scénario, il s’agit de sqoopmerge.

11. Dans la liste File format, sélectionnez le format correspondant aux données utilisées, textfile dans ce scénario.


Configurer le tSqoopImport

Procédure

1. Double-cliquez sur le tSqoopImport afin d’ouvrir sa vue Component.
2. Dans la zone **Mode**, sélectionnez **Use Java API**.

3. Dans la zone **Version**, sélectionnez la distribution Hadoop à utiliser et sa version. Si vous ne trouvez pas la distribution correspondant à la vôtre dans la liste, sélectionnez **Custom** afin de vous connecter à une distribution Hadoop non officiellement supportée par le Studio. Pour un exemple d’utilisation étape par étape de cette option **Custom**, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

4. Dans le champ **NameNode URI**, saisissez l’emplacement du nœud maître, le NameNode, de la distribution utilisée. Par exemple, `hdfs://talend-cdh4-namenode:8020`. Si vous utilisez WebHDFS, l’emplacement doit être `webhdfs://masternode:portnumber`. Si ce WebHDFS est sécurisé via SSL,
le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.


Notez que le mot “Job” dans le terme JobTracker désigne les Jobs MR ou Map/Reduce décrits dans la documentation d’Apache disponible sur le site Internet d’Apache http://hadoop.apache.org/ (en anglais).


Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case Use a keytab to authenticate. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d’accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser.


8. Dans les champs Username et Password, saisissez les informations d’authentification.

9. Sous la table Driver JAR, cliquez sur le bouton [+] pour ajouter une ligne. Dans cette ligne, cliquez sur le bouton [...] pour afficher la liste déroulante et sélectionnez le fichier Jar à utiliser. Dans ce scénario, sélectionnez mysql-connector-java-5.1.30-bin.jar.

Si le bouton [...] n’est pas disponible, cliquez dans la ligne pour le faire apparaître.

10. Dans le champ Table Name, saisissez le nom de la table source. Dans ce scénario, il s’agit de sqoopmerge.

11. Dans la liste File format, sélectionnez le format correspondant aux données utilisées, textfile dans ce scénario.


**Exécuter le Job**

Appuyez sur F6 pour exécuter ce Job.

Une fois l’exécution terminée, vous pouvez vérifier les résultats, dans le dossier cible défini, dans la console Web de la distribution Hadoop utilisée.
Si vous souhaitez obtenir plus de détails sur le Job, il est recommandé d'utiliser la console Web du Jobtracker fourni par la distribution Hadoop utilisée.
tSqoopImportAllTables

Ce composant définit les arguments requis par Sqoop pour écrire toutes les tables d’une base de données dans HDFS.

Le tSqoopImportAllTables appelle Sloop pour transférer toutes les tables d’un système de gestion de base de données relationnelle (SGBDR) comme MySQL ou Oracle dans le système de fichiers distribué Hadoop (Hadoop Distributed File System, HDFS).

Rémarque :
Sqoop est installé dans chaque distribution Hadoop. Cependant, si la distribution Hadoop que vous utilisez n’a pas de Sqoop installé, vous devez en installer un et vous assurer d’ajouter l’invite de commande Sqoop dans la variable PATH de la distribution. Pour plus d’informations concernant l’installation de Sqoop, consultez la documentation de Sqoop.

Propriétés du tSqoopImportAllTables Standard

Ces propriétés sont utilisées pour configurer le tSqoopImportAllTables s’exécutant dans le framework de Jobs Standard.

Le composant tSqoopImportAllTables Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Mode</th>
<th>Sélectionnez le mode dans lequel Sqoop est appelé dans l’exécution du Job.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Commandline</td>
<td>L’interpréteur de commandes Sqoop est utilisé pour appeler Sqoop. Dans ce mode, vous devez déployer et exécuter le Job dans l’hôte où Sqoop est installé. Si votre solution Talend est soumise à souscription, il est recommandé d’installer et d’utiliser un JobServer fourni par Talend dans cet hôte pour exécuter le Job. Si vous utilisez Talend Open Studio for Big Data, assurez-vous que le Studio Talend et Sqoop se trouvent sur la même machine.</td>
</tr>
<tr>
<td>Use Java API</td>
<td>L’API Java est utilisée pour appeler Sqoop. Dans ce mode, le Job peut être exécuté localement dans le Studio Talend mais vous devez configurer la connexion à la distribution Hadoop à utiliser. Notez que JDK est requis pour l’exécution du Job en mode API Java et que les versions des kits JDK installés sur les deux machines doivent être compatibles entre elles. Par exemple, les versions sont les mêmes ou la version JDK de la machine Hadoop est plus récente.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hadoop properties</th>
<th>Peut être Built-in ou Repository :</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>saisissez les informations de configuration de la distribution Hadoop à utiliser localement, pour ce composant uniquement.</td>
</tr>
</tbody>
</table>
• Repository : vous avez déjà créé la connexion Hadoop et l’avez stockée dans le Repository ; vous pouvez la réutiliser directement pour la configuration du composant et la création d’un Job. Pour plus d’informations concernant la création d’une connexion centralisée Hadoop, consultez Guide de prise en main de Talend Open Studio for Big Data.

Distribution

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.


• Si vous sélectionnez Amazon EMR, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

• L’option Custom vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par Talend.

1. Sélectionner Import from existing version pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner Import from zip pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans Talend Exchange, des membres de la Communauté Talend ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste Hadoop configuration (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option Import from existing version, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par Talend. Talend et sa Communauté fournissent l’opportunité de vous
connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d’Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

**Remarque :**

Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d’importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

| Hadoop Version | Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez. |
| NameNode URI | Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée masternode comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé. |
| JobTracker Host | Cochez cette case et, dans le champ qui s’affiche, saisissez l’emplacement du ResourceManager de votre distribution. Par exemple tal-qa114.talend.lan:8050. Cette propriété est requise lorsque la requête que vous souhaitez utiliser est exécutée dans Windows et est une requête Select. Par exemple, SELECT your_column_name FROM your_table_name Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d’un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du de l’exécution) :

1. Cochez la case Set resourcemanager scheduler address et saisissez l’adresse de l’ordonnanceur (Scheduler) dans le champ qui apparait.
3. Cochez la case **Set staging directory** et saisissez le chemin d'accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l'exécution de programmes. Ce répertoire se trouve sous la propriété `yarn.app.mapreduce.am.staging-dir` dans les fichiers de configuration, notamment les fichiers `yarn-site.xml` et `mapred-site.xml` de votre distribution.

4. Allouez des volumes de mémoire aux calculs **Map** et **Reduce** et au service **ApplicationMaster** de YARN en cochant la case **Set memory** dans la vue **Advanced settings**.

5. Cochez la case **Set Hadoop user** et saisissez le nom de l'utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d'écriture, ce champ vous permet d'exécuter le Job directement avec l'utilisateur ayant les droits d'accès appropriés au fichier ou répertoire à traiter.

6. Cochez la case **Use datanode hostname** pour permettre au Job d'accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété `dfs.client.use.datanode.hostname` à `true`. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.


### Use kerberos authentication

Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le 'principal name' de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans [Connexion sécurisée à MapR](#) à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

De plus, comme ce composant effectue des calculs Map/Reduce, vous devez également authentifier les services associés, comme le serveur de l'historique des Jobs et le gestionnaire de ressources ou le JobTracker, selon votre distribution, dans le champ correspondant. Ces principaux se trouvent dans les fichiers de configuration.
de votre distribution. Par exemple, dans une distribution CDH4, le principal du gestionnaire de ressource est configuré dans le fichier `yarn-site.xml` et le principal de l'historique des Job dans le fichier `mapred-site.xml`. Cette case est disponible ou indisponible selon la distribution d'Hadoop à laquelle vous vous connectez.

| Use a keytab to authenticate | Cochez la case Use a keytab to authenticate pour vous connecter à un système utilisant Kerberos à l'aide d'un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ Principal et le chemin d'accès au fichier keytab dans le champ Keytab. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend.

Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est user1 et le principal à utiliser est guest. Dans cette situation, assurez-vous que user1 a les droits de lecture pour le fichier keytab à utiliser.

| Hadoop user name | Saisissez le nom de l'utilisateur avec lequel vous souhaitez exécuter le Job. Puisqu'un fichier ou un répertoire dans Hadoop a son auteur spécifique, avec les droits en lecture ou écriture appropriés, ce champ vous permet d'exécuter le Job directement sous le nom d'utilisateur ayant les droits appropriés pour accéder au fichier ou au répertoire à traiter. Notez que ce champ peut n'être pas disponible selon la distribution que vous utilisez.

| JDBC property | Peut être Built-in ou Repository:

- **Built-in** : saisissez les informations de connexion à la base de données à utiliser localement, pour ce composant uniquement.

- **Repository** : vous devez avoir créé la connexion à la base de données et l'avoir stockée dans le Repository; vous pouvez la réutiliser directement pour la configuration du composant et la création d'un Job. Pour plus d'informations concernant la création d'une connexion à la base de données centralisée, consultez le Guide utilisateur du Studio Talend.

Notez que seules les connexions de type General JDBC stockées dans le Repository sont supportées.

| Connection | Saisissez l'URL JDBC utilisée pour vous connecter à la base de données dans laquelle les données source sont stockées.

| User name et Password | Saisissez les informations d'authentification utilisées pour vous connecter à la base de données source. |
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres.

Si votre mot de passe est stocké dans un fichier, cochez la case **The password is stored in a file** et saisissez le chemin d’accès à ce fichier dans le champ **File path** qui s’affiche.

- Ce fichier peut être stocké sur la machine où le Job est exécuté ou dans le système HDFS du cluster Hadoop à utiliser.
- Le mot de passe stocké dans ce fichier ne doit pas contenir “\n” (nouvelle ligne) à la fin. Vous ne devez pas insérer de nouvelle ligne, même vide, à la fin du mot de passe.

### Driver JAR

Quel que soit le mode, **Use Commandline** ou **Java API**, vous devez ajouter le fichier du pilote de la base de données à utiliser, dans le dossier **lib** de la distribution Hadoop que vous utilisez. Pour ce faire, utilisez la table **Driver JAR** afin d’ajouter ce fichier de pilote pour le Job en cours de construction.

### File format

Sélectionnez un format de fichier pour les données à transférer:
- **textfile**
- **sequencefile**
- **Avro file**
- **Parquet file**: la version de Sqoop doit être 1.4.6.

### Specify Number of Mappers

Cochez cette case afin de préciser le nombre de tâches de "map" (processus parallèles) utilisé pour effectuer le transfert des données.

Si ne vous souhaitez pas que Sqoop travaille en parallèle, saisissez 1 dans le champ affiché.

### Compress

Cochez cette case pour activer la compression.

### Direct

Cochez cette case pour utiliser le chemin d’import rapide.

### Exclude table

Cochez cette case et saisissez le nom de(s) la table(s) à exclure du processus d’import.

### Print Log

Cochez cette case pour activer la case **Verbose**.

### Verbose

Cochez cette case pour imprimer plus d’informations pendant que vous travaillez, par exemple des informations de débogage.
## Advanced settings

<table>
<thead>
<tr>
<th>Define Hive mapping</th>
<th>Cette fonctionnalité est disponible selon la version de Sqoop que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use MySQL default delimiters</td>
<td>Cochez cette case pour utiliser les séparateurs MySQL par défaut. Cette case est disponible en mode Commandline (Use Commandline).</td>
</tr>
<tr>
<td>Additional arguments</td>
<td>Complétez ce tableau pour utiliser des arguments supplémentaires, si nécessaire. En ajoutant des arguments supplémentaires, vous pouvez effectuer de multiples opérations en une seule transaction. Par exemple, vous pouvez utiliser <code>--hive-import</code> et <code>--hive-table</code> en mode Commandline (Use Commandline) ou <code>hive.import</code> et <code>hive.table.name</code> en mode Java API (Use Java API) pour créer une table Hive et écrire des données dans cette table lors de l'exécution de la transaction écrivant des données dans HDFS. Pour plus d'informations concernant les arguments Sqoop disponibles en mode Commandline et en mode Java API, consultez Arguments supplémentaires à la page 3907.</td>
</tr>
<tr>
<td>Hadoop properties</td>
<td>Le Studio Talend utilise une configuration par défaut pour son moteur, afin d'effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l'exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut. • Notez que, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu'à ce que, dans la liste Property type, vous passiez de Repository à Built-in. Pour plus d'informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d'Apache Hadoop sur <a href="http://hadoop.apache.org/docs">http://hadoop.apache.org/docs</a> en sélectionnant la version de la documentation souhaitée. À titre d'exemple, les liens vers certaines propriétés sont listés ci-après : • Généralement, les propriétés relatives à HDFS peuvent être trouvées dans le fichier <code>hdfs-default.xml</code> correspondant à votre distribution, comme par exemple <a href="http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml">http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml</a> (en anglais). • Apache fournit également une page listant toutes les propriétés relatives à Hive : <a href="http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml">http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml</a> (en anglais).</td>
</tr>
</tbody>
</table>
Vous pouvez personnaliser les opérations map et reduce en cochant la case **Set memory**, pour configurer les allocations de mémoire pour ces opérations à effectuer par le système Hadoop.

Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations *map* et *reduce* dans les champs **Mapred job map memory mb** et **Mapred job reduce memory mb**, respectivement. Par défaut, les valeurs sont toutes les deux 1000, ce qui est normalement adapté pour l’exécution de ces opérations.

Les paramètres de mémoire à définir sont **Map (in Mb)**, **Reduce (in Mb)** et **ApplicationMaster (in Mb)**. Ces champs permettent d’allouer dynamiquement de la mémoire aux opérations *map* et *reduce* et à l’ApplicationMaster de YARN.

Laissez le champ **Path separator in server** tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d’autres termes, changez le séparateur si celui-ci n’est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte.

Cochez cette case pour collecter les données de log au niveau des composants.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

**EXIT_CODE** : code de sortie de la commande distante. Cette variable est une variable *After* et retourne un nombre entier.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
Il est recommandé d’utiliser une version 1.4+ de Sqoop afin de tirer entièrement parti des fonctionnalités des composants.

Pour plus d’informations concernant Sqoop, consultez (en anglais) le manuel de Sqoop sur le site : http://sqoop.apache.org/docs/.

### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend. La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native. Par exemple, pour Windows, la bibliothèque est lib\MapRClient.dll dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr (en anglais).

  Si vous n’ajoutez pas de librairie, il est possible que vous rencontriez l’erreur suivante : no MapRClient in java.library.path.

- Configurez l’argument -Djava.library.path, par exemple, dans la zone Job Run VM arguments de la vue Run/Debug de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

  Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

### Limitation

Si vous avez sélectionné le mode Use Commandline, vous devez utiliser l’hôte où Sqoop est installé pour exécuter le Job à l’aide de ce composant.

Les prérequis de Sqoop pour l’utilisation de son outil import-all-tables doivent être respectés. Pour plus d’informations, consultez le manuel de Sqoop.

### Connections

Liens de sortie (de ce composant à un autre) :

**Trigger** : Run if, On Subjob Ok, On Subjob Error.

Liens d’entrée (d’un autre composant à celui-ci) :

**Row** : Iterate.

Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tSqoopMerge

Ce composant effectue un import incrémental mettant à jour un ancien ensemble de données avec des enregistrements plus récents. Le type des fichiers des deux ensembles de données doit être le même.

Le tSqoopMerge lit deux ensembles de données dans HDFS et les combine à l'aide d’une classe de fusion (merge) capable d’analyser des ensembles de données. Les nouveaux enregistrements écrasent les anciens.

Remarque :
Sqoop est installé dans chaque distribution Hadoop. Cependant, si la distribution Hadoop que vous utilisez n’a pas de Sqoop installé, vous devez en installer un et vous assurer d’ajouter l’invite de commande Sqoop dans la variable PATH de la distribution. Pour plus d’informations concernant l’installation de Sqoop, consultez la documentation de Sqoop.

Propriétés du tSqoopMerge Standard

Ces propriétés sont utilisées pour configurer le tSqoopMerge s’exécutant dans le framework de Jobs Standard.

Le composant tSqoopMerge Standard appartient aux familles Big Data et File.

Le composant de ce framework est disponible lorsque vous utilisez l’une des solutions Big Data de Talend.

Basic settings

<table>
<thead>
<tr>
<th>Mode</th>
<th>Sélectionnez le mode dans lequel Sqoop est appelé dans l’exécution du Job.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Commandline</td>
<td>l’interpréteur de commandes Sqoop est utilisé pour appeler Sqoop. Dans ce mode, vous devez déployer et exécuter le Job dans l’hôte où Sqoop est installé. Si votre solution Talend est soumise à souscription, il est recommandé d’installer et d’utiliser un JobServer fourni par Talend dans cet hôte pour exécuter le Job. Si vous utilisez Talend Open Studio for Big Data, assurez-vous que le Studio Talend et Sqoop se trouvent sur la même machine.</td>
</tr>
<tr>
<td>Use Java API</td>
<td>l’API Java est utilisée pour appeler Sqoop. Dans ce mode, le Job peut être exécuté localement dans le Studio Talend mais vous devez configurer la connexion à la distribution Hadoop à utiliser. Notez que JDK est requis pour l’exécution du Job en mode API Java et que les versions des kits JDK installés sur les deux machines doivent être compatibles entre elles. Par exemple, les versions sont les mêmes ou la version JDK de la machine Hadoop est plus récente.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hadoop properties</th>
<th>Peut être Built-in ou Repository :</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Built-in</td>
<td>saisissez les informations de configuration de la distribution Hadoop à utiliser localement, pour ce composant uniquement.</td>
</tr>
</tbody>
</table>
### Repository

- **Repository** : vous avez déjà créé la connexion Hadoop et l’avez stockée dans le **Repository** ; vous pouvez la réutiliser directement pour la configuration du composant et la création d’un Job. Pour plus d’informations concernant la création d’une connexion centralisée Hadoop, consultez *Guide de prise en main de Talend Open Studio for Big Data*.

---

### Distribution

Sélectionnez dans la liste le cluster que vous utilisez. Les options de la liste varient selon le composant que vous utilisez. Les options de la liste dépendent des composants que vous utilisez, Parmi ces options, les suivantes nécessitent une configuration spécifique.


- Si vous sélectionnez **Amazon EMR**, vous pouvez trouver plus d’informations concernant la configuration d’un cluster Amazon EMR dans Talend Help Center (https://help.talend.com).

- L’option **Custom** vous permet de vous connecter à un cluster différent des clusters de la liste, par exemple une distribution non supportée officiellement par **Talend**.

1. Sélectionner **Import from existing version** pour importer une distribution de base officiellement supportée et ajouter manuellement les autres Jars requis non fournis par cette distribution. Ou,

2. Sélectionner **Import from zip** pour importer le fichier .zip de configuration pour la distribution personnalisée à utiliser. Ce fichier zip doit contenir les bibliothèques des différents éléments Hadoop et le fichier d’index de ces bibliothèques.

Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** (en anglais) et utiliser directement dans votre connexion. Cependant, avec l’évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d’utiliser l’option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend. Talend** et sa Communauté fournissent l’opportunité de vous.
connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d'Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

Remarque :
Dans cette boîte de dialogue, la case de la zone active doit être cochée, afin d'importer les fichiers .jar correspondant à la connexion créée entre la distribution personnalisée et ce composant.

Pour un exemple étape par étape expliquant comment se connecter à une distribution personnalisée et partager cette connexion, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.

<table>
<thead>
<tr>
<th>Hadoop Version</th>
<th>Sélectionnez la version de la distribution Hadoop que vous utilisez. Les options disponibles dépendent du composant que vous utilisez.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameNode URI</td>
<td>Saisissez l’URI du NameNode Hadoop, nœud maître d’un système Hadoop. Par exemple, si vous avez sélectionné une machine nommée masternode comme NameNode, son emplacement est hdfs://masternode:portnumber. Si vous utilisez WebHDFS, l’emplacement doit être webhdfs://masternode:portnumber. Si ce WebHDFS est sécurisé via SSL, le schéma d’URI doit être swebhdfs et vous devez utiliser un tLibraryLoad dans le Job pour charger la bibliothèque requise par votre WebHDFS sécurisé.</td>
</tr>
</tbody>
</table>
| JobTracker Host | Cochez cette case et, dans le champ qui s’affiche, saisissez l’emplacement du ResourceManager de votre distribution. Par exemple tal-qa114.tale nd.lan:8050. Cette propriété est requise lorsque la requête que vous souhaitez utiliser est exécutée dans Windows et est une requête Select. Par exemple, SELECT your_column_name FROM your_table_name Vous pouvez continuer à configurer les paramètres suivants selon la configuration du cluster Hadoop à utiliser (si vous ne cochez pas la case d’un paramètre, alors la configuration de ce paramètre dans le cluster Hadoop à utiliser sera ignorée lors du de l’exécution) :

1. Cochez la case Set resourcemanager scheduler address et saisissez l’adresse de l’ordonnanceur (Scheduler) dans le champ qui apparaît.

3. Cochez la case **Set staging directory** et saisissez le chemin d’accès au répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créés par l’exécution de programmes. Ce répertoire se trouve sous la propriété `yarn.app.mapreduce.am.staging-dir` dans les fichiers de configuration, notamment les fichiers `yarn-site.xml` et `mapred-site.xml` de votre distribution.

4. Allouez des volumes de mémoire aux calculs **Map** et **Reduce** et au service **ApplicationMaster** de YARN en cochant la case **Set memory** dans la vue **Advanced settings**.

5. Cochez la case **Set Hadoop user** et saisissez le nom de l’utilisateur avec lequel vous souhaitez exécuter le Job. Puisque les fichiers et répertoires dans Hadoop ont un auteur spécifique avec les droits appropriés de lecture ou d’écriture, ce champ vous permet d’exécuter le Job directement avec l’utilisateur ayant les droits d’accès appropriés au fichier ou répertoire à traiter.

6. Cochez la case **Use datanode hostname** pour permettre au Job d’accéder aux nœuds de données via leurs hébergeurs. Cela configure la propriété `dfs.client.use.datanode.hostname` à `true`. Lorsque vous vous connectez à un système de fichiers S3N, vous devez cocher cette case.


### Use kerberos authentication

Si vous accédez au cluster Hadoop fonctionnant avec la sécurité de Kerberos, cochez cette case, puis saisissez le ’principal name’ de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d’utiliser votre identifiant pour vous authentifier, en le comparant aux identifiants stockés dans Kerberos.

- Si ce cluster est un cluster MapR de version 5.0.0 ou postérieure, vous pouvez paramétrer la configuration de l’authentification par ticket MapR en plus ou comme une alternative en suivant les explications dans **Connexion sécurisée à MapR** à la page 1745.

Gardez à l’esprit que cette configuration génère un nouveau ticket de sécurité MapR pour le nom d’utilisateur défini dans le Job dans chaque exécution. Si vous devez réutiliser un ticket existant provenant du même utilisateur, laissez décochées les cases **Force MapR ticket authentication** et **Use Kerberos authentication**. MapR devrait pouvoir trouver automatiquement ce ticket à la volée.

De plus, comme ce composant effectue des calculs Map/Reduce, vous devez également authentifier les services associés, comme le serveur de l’historique des Jobs et le gestionnaire de ressources ou le JobTracker, selon votre distribution, dans le champ correspondant. Ces principaux se trouvent dans les fichiers de configuration...
De votre distribution. Par exemple, dans une distribution CDH4, le principal du gestionnaire de ressource est configuré dans le fichier `yarn-site.xml` et le principal de l'historique des Job dans le fichier `mapred-site.xml`. Cette case est disponible ou indisponible selon la distribution d’Hadoop à laquelle vous vous connectez.

<table>
<thead>
<tr>
<th>Use a keytab to authenticate</th>
<th>Cochez la case <strong>Use a keytab to authenticate</strong> pour vous connecter à un système utilisant Kerberos à l'aide d’un fichier keytab. Un fichier keytab contient des paires de principaux Kerberos et de clés cryptées. Vous devez saisir le principal à utiliser dans le champ <strong>Principal</strong> et le chemin d'accès au fichier keytab dans le champ <strong>Keytab</strong>. Ce fichier keytab doit être stocké sur la machine où s'exécute votre Job, par exemple, sur un serveur de Jobs Talend. Notez que l'utilisateur qui exécute un Job utilisant un keytab n'est pas forcément celui désigné par le principal mais qu'il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter le Job est <strong>user1</strong> et le principal à utiliser est <strong>guest</strong>. Dans cette situation, assurez-vous que <strong>user1</strong> a les droits de lecture pour le fichier keytab à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hadoop user name</strong></td>
<td>Saisissez le nom de l'utilisateur avec lequel vous souhaitez exécuter le Job. Puisqu’un fichier ou un répertoire dans Hadoop a son auteur spécifique, avec les droits en lecture ou écriture appropriés, ce champ vous permet d'exécuter le Job directement sous le nom d'utilisateur ayant les droits appropriés pour accéder au fichier ou au répertoire à traiter. Notez que ce champ peut n'être pas disponible selon la distribution que vous utilisez.</td>
</tr>
<tr>
<td><strong>Old data directory</strong></td>
<td>Saisissez le chemin d'accès à l'ancien ensemble de données à fusionner.</td>
</tr>
<tr>
<td><strong>New data directory</strong></td>
<td>Saisissez le chemin d'accès au nouvel ensemble de données à fusionner.</td>
</tr>
<tr>
<td><strong>Target directory</strong></td>
<td>Saisissez le chemin du répertoire dans lequel écrire en sortie le résultat de la fusion.</td>
</tr>
<tr>
<td><strong>Merge key</strong></td>
<td>Saisissez le nom de la colonne utilisée comme clé de chaque enregistrement pour la fusion. La clé primaire doit être unique.</td>
</tr>
<tr>
<td><strong>Need to generate the JAR file</strong></td>
<td>Cochez cette case pour générer le fichier jar de fusion et la classe de fusion (merge) requis pour analyser les ensembles de données à fusionner. Le nom par défaut du fichier jar et de la classe est <code>SqoopMerge_component_ID</code>. Ce <code>component_ID</code> est l'identifiant du composant <code>tSqoopMerge</code> qui génère le fichier jar et la classe, par exemple <code>tSqoopMerge_1</code> ou <code>tSqoopMerge_2</code>.</td>
</tr>
<tr>
<td>JDBC property</td>
<td>Ce fichier Jar est généré de la table source des données importées. Cocher cette case permet d’afficher les paramètres à configurer pour se connecter à la table.</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>JDBC property</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td>• <strong>Built-in</strong> : saisissez les informations de connexion à la base de données à utiliser localement, pour ce composant uniquement.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Repository</strong> : vous devez avoir créé la connexion à la base de données et l’avoir stockée dans le Repository; vous pouvez la réutiliser directement pour la configuration du composant et la création d’un Job. Pour plus d’informations concernant la création d’une connexion à la base de données centralisée, consultez le Guide utilisateur du Studio Talend. Notez que seules les connexions de type <strong>General JDBC</strong> stockées dans le Repository sont supportées.</td>
<td></td>
</tr>
<tr>
<td><strong>Connection</strong></td>
<td>Saisissez l’URL JDBC utilisée pour vous connecter à la base de données dans laquelle les données source sont stockées.</td>
</tr>
<tr>
<td><strong>User name et Password</strong></td>
<td>Saisissez les informations d’authentification utilisées pour vous connecter à la base de données source.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table Name</strong></td>
<td>Saisissez le nom de la table source.</td>
</tr>
<tr>
<td></td>
<td>Ce nom est utilisé pour nommer le fichier Jar généré.</td>
</tr>
<tr>
<td><strong>Driver JAR</strong></td>
<td>Quel que soit le mode, <strong>Use Commandline</strong> ou <strong>Java API</strong>, vous devez ajouter le fichier du pilote de la base de données à utiliser, dans le dossier <strong>lib</strong> de la distribution Hadoop que vous utilisez. Pour ce faire, utilisez la table Driver JAR afin d’ajouter ce fichier de pilote pour le Job en cours de construction. Le Jar de ce pilote est requis uniquement lorsque vous devez vous connecter à la base de données à utiliser pour générer le fichier Jar de fusion. Cette table <strong>Driver JAR</strong> est disponible uniquement lorsque vous avez coché la case <strong>Need to generate the JAR file</strong>.</td>
</tr>
<tr>
<td><strong>JAR file</strong></td>
<td>Si la classe de fusion (merge) existe déjà et est disponible, saisissez le chemin d’accès vers le fichier jar qui contient cette classe afin de la réutiliser.</td>
</tr>
<tr>
<td></td>
<td>Dans ce cas, vous devez saisir le nom de la classe dans le champ <strong>Class name</strong> de l’onglet <strong>Advanced settings</strong>.</td>
</tr>
<tr>
<td><strong>Print Log</strong></td>
<td>Cochez cette case pour activer la case <strong>Verbose</strong>.</td>
</tr>
</tbody>
</table>
### Verbose

Cochez cette case pour imprimer plus d’informations pendant que vous travaillez, par exemple des informations de débogage.

### Advanced settings

#### Custom class name

Cochez cette case pour afficher le champ `Class name` et saisissez le nom de la classe de fusion (merge) que vous souhaitez utiliser.

Cette case doit être décochée si vous souhaitez utiliser l’option Generate the JAR file dans l’onglet Basic settings.

#### Additional arguments

Complétez ce tableau pour utiliser des arguments supplémentaires, si nécessaire.

En ajoutant des arguments supplémentaires, vous pouvez effectuer de multiples opérations en une seule transaction. Par exemple, vous pouvez utiliser `--hive-import` et `--hive-table en mode Commandline (Use Commandline)` ou `hive.import` et `hive.table.name en mode Java API (Use Java API)` pour créer une table Hive et écrire des données dans cette table lors de l’exécution de la transaction écrivant des données dans HDFS. Pour plus d’informations concernant les arguments Sqoop disponibles en mode Commandline et en mode Java API, consultez Arguments supplémentaires à la page 3907.

#### Hadoop properties

Le Studio Talend utilise une configuration par défaut pour son moteur, afin d’effectuer des opérations dans une distribution Hadoop. Si vous devez utiliser une configuration personnalisée dans une situation spécifique, renseignez dans cette table la ou les propriété(s) à personnaliser. Lors de l’exécution, la ou les propriété(s) personnalisée(s) va (vont) écraser celle(s) par défaut.

- **Notez que**, si vous utilisez les métadonnées stockées centralement dans le Repository, cette table hérite automatiquement des propriétés définies dans ces métadonnées et passe en lecture seule jusqu’à ce que, dans la liste Property type, vous passiez de Repository à Built-in.

Pour plus d’informations concernant les propriétés requises par Hadoop et ses systèmes associés, tels que HDFS et Hive, consultez la documentation de la distribution Hadoop utilisée ou consultez la documentation d’Apache Hadoop sur [http://hadoop.apache.org/docs](http://hadoop.apache.org/docs) en sélectionnant la version de la documentation souhaitée. À titre d’exemple, les liens vers certaines propriétés sont listés ci-après :

- Généralement, les propriétés relatives à HDFS peuvent être trouvées dans le fichier `hdfs-default.xml` correspondant à votre distribution, comme par exemple [http://hadoop.apache.org/docs/r2.6.0/](http://hadoop.apache.org/docs/r2.6.0/)

| **Mapred job map memory mb et Mapred job reduce memory mb** | Vous pouvez personnaliser les opérations map et reduce en cochant la case Set memory, pour configurer les allocations de mémoire pour ces opérations à effectuer par le système Hadoop.

Dans ce cas, vous devez saisir les valeurs que vous souhaitez utiliser pour la mémoire allouée aux opérations map et reduce dans les champs Mapred job map memory mb et Mapred job reduce memory mb, respectivement. Par défaut, les valeurs sont toutes les deux 1000, ce qui est normalement adapté pour l’exécution de ces opérations.

Les paramètres de mémoire à définir sont Map (in Mb), Reduce (in Mb) et ApplicationMaster (in Mb). Ces champs permettent d’allouer dynamiquement de la mémoire aux opérations map et reduce et à l’ApplicationMaster de YARN. |
| **Path separator in server** | Laissez le champ Path separator in server tel quel, sauf si vous changez le séparateur utilisé par la machine hôte de votre distribution Hadoop pour sa variable PATH. En d’autres termes, changez le séparateur si celui-ci n’est pas le signe deux points (:). Dans ce cas, vous devez remplacer cette valeur par celle utilisée dans votre hôte. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau des composants. |

### Variables globales

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

**EXIT_CODE** : code de sortie de la commande distante. Cette variable est une variable After et retourne un nombre entier.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
## Utilisation

### Règle d’utilisation


Il est recommandé d’utiliser une version 1.4+ de Sqoop afin de tirer entièrement parti des fonctionnalités des composants.


### Prérequis

La distribution Hadoop doit être correctement installée afin de garantir les interactions avec le Studio Talend.

La liste suivante présente des informations d’exemple relatives à MapR.

- Assurez-vous d’avoir installé le client MapR sur la même machine que le Studio Talend et d’avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D’après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées `MAPR_INSTALL\hadoop\hadoop-VERSION\lib\native`. Par exemple, pour Windows, la bibliothèque `lib\MapRClient.dll` dans le fichier Jar du client MapR. Pour plus d’informations, consultez la page suivante sur le site de MapR : [http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr](http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr) (en anglais).

Si vous n’ajoutez pas de librairie, il est possible que vous rencontrez l’erreur suivante : `no MapRClient in java.library.path`.

- Configurez l’argument `-Djava.library.path`, par exemple, dans la zone **Job Run VM arguments** de la vue **Run/Debug** de la boîte de dialogue [Preferences] dans le menu Window. Cet argument fournit au studio le chemin d’accès à la bibliothèque native du client MapR. Cela permet aux utilisateurs en souscription d’utiliser entièrement l’aperçu des données (Data viewer) afin de visualiser localement dans le studio les données stockées dans MapR.

Pour plus d’informations concernant l’installation d’une distribution Hadoop, consultez le manuel correspondant à la distribution Hadoop que vous utilisez.

### Limitation

Si vous avez sélectionné le mode **Use Commandline**, vous devez utiliser l’hôte où Sqoop est installé pour exécuter le Job à l’aide de ce composant.

### Connections

Liens de sortie (de ce composant à un autre) :


Liens d’entrée (d’un autre composant à celui-ci) :

**Row** : Iterate.
Scénario : Fusionner deux ensembles de données dans HDFS

Ce scénario s’applique uniquement aux solutions Talend avec Big Data.

Ce scénario montre comment utiliser le tSqoopMerge pour fusionner deux ensembles de données importés de manière séquentielle dans HDFS, à partir de la même table MySQL, mais dont un enregistrement a été modifié.

Le premier ensemble de données utilisé (celui avant les modifications) se présente comme suit :

<table>
<thead>
<tr>
<th>id, wage, mod_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 2000, 2008-06-26 04:25:59</td>
</tr>
<tr>
<td>1, 2300, 2011-06-12 05:29:45</td>
</tr>
<tr>
<td>3, 3000, 2010-05-02 15:34:05</td>
</tr>
</tbody>
</table>

Le chemin vers cet ensemble de données dans HDFS est /user/ychen/target_old.

Le second ensemble de données utilisé (celui après les modifications) se présente comme suit :

<table>
<thead>
<tr>
<th>id, wage, mod_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 2000, 2008-06-26 04:25:59</td>
</tr>
<tr>
<td>1, 2300, 2011-06-12 05:29:45</td>
</tr>
<tr>
<td>3, 4000, 2013-10-14 18:00:00</td>
</tr>
</tbody>
</table>

Le chemin vers cet ensemble de données dans HDFS est /user/ychen/target_new.

Ces ensembles de données sont tous les deux importés à l’aide du composant tSqoopImport. Pour un scénario montrant comment utiliser le tSqoopImport, consultez Scénario : Importer une table MySQL dans HDFS à la page 3929.

Le Job décrit dans ce scénario fusionne deux ensembles de données. Les nouveaux enregistrements écrasent les anciens.

Avant de commencer à reproduire ce scénario, vérifiez que vous disposez des droits et permissions nécessaires pour accéder à la distribution Hadoop utilisée. Ensuite, procédez comme suit :


Pour plus d’informations concernant les connexions, consultez le Guide utilisateur du Studio Talend.
**Déposer le composant**

**Procédure**

1. Dans la perspective *Integration* du Studio, créez un Job vierge depuis le nœud *Job Designs* du *Repository*.
   Pour plus d’informations sur la création d’un Job, consultez le *Guide utilisateur du Studio Talend*.

2. Déposez un *tSqoopMerge* dans l’espace de modélisation graphique.
   Dans ce scénario, le fichier .jar nécessaire pour la fusion n’est pas disponible. Vous devez donc utiliser un *tSqoopMerge* afin de la générer à l’exécution à partir de la table MySQL source.
   Pour plus d’informations sur le *tLibraryLoad*, consultez *tLibraryLoad* à la page 2083.

**Configurer le tSqoopMerge**

**Procédure**

1. Double-cliquez sur le *tSqoopMerge* afin d’ouvrir sa vue *Component*. 
2. Dans la zone **Mode**, sélectionnez **Use Java API**.

3. Dans la zone **Version**, sélectionnez la distribution Hadoop à utiliser et sa version. Si vous ne trouvez pas la distribution correspondant à la votre dans la liste, sélectionnez **Custom** afin de vous connecter à une distribution Hadoop non officiellement supportée par le Studio.

   Pour un exemple d’utilisation étape par étape de cette option **Custom**, consultez Connexion à une distribution Hadoop personnalisée à la page 1677.


6. Si la distribution utilisée nécessite une authentification Kerberos, cochez la case **Use Kerberos authentication** et saisissez les informations de connexion. Sinon, laissez cette case décochée.

   Si vous souhaitez utiliser un fichier Kerberos keytab pour vous identifier, cochez la case **Use a keytab to authenticate**. Un fichier keytab contient des paires de principaux et clés cryptées Kerberos. Vous devez saisir le principal à utiliser dans le champ **Principal** et le chemin d’accès au fichier keytab dans le champ **Keytab**. Ce fichier keytab doit être stocké sur la machine où s’exécute votre Job, par exemple, sur un serveur de Jobs Talend.

   Notez que l’utilisateur qui exécute un Job utilisant un keytab n’est pas forcément celui désigné par le principal mais qu’il doit avoir le droit de lecture pour le fichier keytab utilisé. Par exemple, le nom d’utilisateur que vous utilisez pour exécuter le Job est **user1** et le principal à utiliser est **guest**. Dans cette situation, assurez-vous que **user1** a les droits de lecture pour le fichier keytab à utiliser.

7. Dans les champs **Old data directory** et **New data directory**, saisissez le chemin ou parcourez votre système de fichiers vers le dossier contenant respectivement l’ancien et le nouvel ensemble de données dans HDFS.

8. Dans le champ **Target directory**, saisissez le chemin ou parcourez votre système de fichiers vers le dossier contenant les données fusionnées.

9. Dans le champ **Merge key**, saisissez le nom de la colonne utilisée en tant que clé pour la fusion. Dans ce scénario, cette colonne est **id**.

10. Cochez la case **Need to generate the JAR file** afin d’afficher les paramètres de connexion de la table de base de données source.

11. Dans le champ **Connection**, saisissez l’URI de la base de données MySQL contenant la table source. Par exemple, **jdbc:mysql://10.42.10.13/mysql**.

12. Dans le champ **Table Name**, saisissez le nom de la table source, **sqoopmerge** dans ce scénario.


14. Sous la table **Driver JAR**, cliquez sur le bouton [+] pour afficher la liste déroulante et sélectionner le fichier Jar à utiliser. Dans ce scénario, sélectionnez **mysql-connector-java-5.1.30-bin.jar**.

   Si le bouton [...] n’est pas disponible, cliquez dans la ligne pour le faire apparaître.

15. Si le séparateur de champs de la table source n’est pas la virgule (,), vous devez le définir dans la table **Additional Arguments** de la vue **Advanced settings**. L’argument utilisé est **codegen.output .delimiters.field** pour le mode **Use Java API** ou **--fields-terminated-by** pour le mode **Use Commandline**.

**Exécuter le Job**

Appuyez sur **F6** pour exécuter ce Job.

Pendant l’exécution, le fichier .jar et la classe pour la fusion sont générés sur la machine locale.
Une fois l’exécution terminée, vous pouvez vérifier les résultats, dans le dossier cible défini, dans la console Web de la distribution Hadoop utilisée.

**File: /user/ychen/merged/part-r-00000**

Si vous souhaitez obtenir plus de détails sur le Job, il est recommandé d’utiliser la console Web du Jobtracker fournie par la distribution Hadoop utilisée.

Si vous continuez à importer des ensembles de données dans HDFS à partir de la même table source, vous pouvez réutiliser la classe de fusion (merge) générée pour fusionner des ensembles de données.
**tSQSCConnection**

Ce composant ouvre une connexion à Amazon Simple Queue Service pouvant être réutilisée par d'autres composants SQS.

### Propriétés du tSQSCConnection standard

Ces propriétés sont utilisées pour configurer le tSQSCConnection s'exécutant dans le framework des Jobs standard.

Le composant tSQSCConnection standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

#### Basic settings

<table>
<thead>
<tr>
<th>Access Key et Secret Key</th>
<th>Spécifiez les clés d'accès (l'ID de la clé d'accès dans le champ <strong>Access Key</strong> et la clé secrète d'accès dans le champ <strong>Secret Key</strong>) requises pour accéder à Amazon Web Services. Pour plus d'informations concernant les clés d'accès d'AWS, consultez Clés d'accès (ID de clé d'accès et clé d'accès secrète). Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ <strong>Secret key</strong>, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour tirer parti des informations d'authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
</tbody>
</table>
| Assume Role | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  
- **Role ARN** : nom Amazon Resource Name (ARN) du rôle.  
- **Role session name** : identifiant de la session du rôle.  
- **Session duration (minutes)** : durée (en minutes) pour laquelle est active la session du rôle.  
Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais). |
| Region | Spécifiez la région AWS en sélectionnant un nom de région ou en saisissant une région entre guillemets doubles (par exemple `'us-east-1'`) dans la liste. Pour plus |
d’informations concernant les régions AWS, consultez Régions et points de terminaison AWS.

**Advanced settings**

**Config client**
Cochez cette case et, dans la table affichée, spécifiez les paramètres de configuration de la table. Cliquez sur le bouton [+] sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre de configuration client et configurez la valeur des attributs suivants pour chaque paramètre :
- **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.
- **Value** : saisissez la valeur du paramètre correspondant.

**STS Endpoint**
Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.
Cette case est disponible uniquement lorsque la case **Assume role** est cochée.

**tStatCatcher Statistics**
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

**Variables globales**

**ERROR_MESSAGE**
Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

**Utilisation**

Règle d’utilisation
Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job.

**Scénarios associés**

- Livrer des messages dans une file Amazon SQS à la page 3980
- Lister des files Amazon SQS dans une région AWS à la page 3997
- Récupérer des messages d’une file Amazon SQS à la page 3965
tSQSInput

Ce composant récupère un ou plusieurs messages, dans la limite de dix messages, depuis une file Amazon SQS (Simple Queue Service).

**Propriétés du tSQSInput standard**

Ces propriétés sont utilisées pour configurer le tSQSInput s’exécutant dans le framework des Jobs standard.

Le composant tSQSInput standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Key et Secret Key</td>
<td>Spécifiez les clés d’accès (l’ID de la clé d’accès dans le champ Access Key et la clé secrète d’accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés d’accès d’AWS, consultez Clés d’accès (ID de clé d’accès et clé d’accès secrète). Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Access Key et Secret Key</td>
<td>Spécifiez les clés d’accès (l’ID de la clé d’accès dans le champ Access Key et la clé secrète d’accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés</td>
</tr>
</tbody>
</table>
### tSQSInput

<table>
<thead>
<tr>
<th>Commentaire</th>
<th>Texte</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d'accès d'AWS, consultez Clés d'accès (ID de clé d'accès et clé d'accès secrète).</td>
<td></td>
<td>Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ <strong>Secret key</strong>, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Inherit credentials from AWS role</strong></td>
<td></td>
<td>Cochez cette case pour tirer parti des informations d'authentification du profil de l'instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d'Amazon EC2. Pour utiliser cette option, votre Job doit s'exécuter dans Amazon EC2 ou d'autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d'informations, consultez Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2.</td>
</tr>
<tr>
<td><strong>Assume Role</strong></td>
<td></td>
<td>Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.</td>
</tr>
<tr>
<td>• <strong>Role ARN</strong> : nom Amazon Resource Name (ARN) du rôle.</td>
<td></td>
<td>• <strong>Role session name</strong> : identifiant de la session du rôle.</td>
</tr>
<tr>
<td>• <strong>Session duration (minutes)</strong> : durée (en minutes) pour laquelle est active la session du rôle.</td>
<td></td>
<td>• <strong>Session duration (minutes)</strong> : durée (en minutes) pour laquelle est active la session du rôle.</td>
</tr>
<tr>
<td>Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).</td>
<td></td>
<td>Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).</td>
</tr>
<tr>
<td><strong>Region</strong></td>
<td></td>
<td>Spécifiez la région AWS en sélectionnant un nom de région ou en saisissant une région entre guillemets doubles (par exemple <strong>us-east-1</strong>) dans la liste. Pour plus d’informations concernant les régions AWS, consultez Régions et points de terminaison AWS.</td>
</tr>
<tr>
<td><strong>Queue (Name or URL)</strong></td>
<td></td>
<td>Spécifiez le nom ou l’URL de la file de laquelle récupérer les messages.</td>
</tr>
<tr>
<td><strong>Schema et Edit schema</strong></td>
<td></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs.</td>
</tr>
<tr>
<td>Le schéma de ce composant est prédéfini, avec les quatre attributs de messages de type String suivants :</td>
<td></td>
<td>Le schéma de ce composant est prédéfini, avec les quatre attributs de messages de type String suivants :</td>
</tr>
<tr>
<td>• <strong>MessageId</strong> : ID du message.</td>
<td></td>
<td>• <strong>MessageId</strong> : ID du message.</td>
</tr>
<tr>
<td>• <strong>MD5OfBody</strong> : MD5 du corps du message.</td>
<td></td>
<td>• <strong>MD5OfBody</strong> : MD5 du corps du message.</td>
</tr>
<tr>
<td>• <strong>Body</strong> : corps du message.</td>
<td></td>
<td>• <strong>Body</strong> : corps du message.</td>
</tr>
<tr>
<td>Vous pouvez cliquer sur le bouton à côté du champ <strong>Edit schema</strong> pour voir son schéma et le modifier en</td>
<td></td>
<td>Vous pouvez cliquer sur le bouton à côté du champ <strong>Edit schema</strong> pour voir son schéma et le modifier en</td>
</tr>
</tbody>
</table>
supprimant les attributs existant ou en ajoutant des attributs SQS standard et des attributs personnalisés.

**Read standard attributes**
Cochez cette case pour récupérer les attributs Amazon SQS standard avec chaque message. Pour plus d’informations concernant les attributs Amazon SQS, consultez [ReceiveMessage > Request Parameters](#) (en anglais).

**Read custom user attributes**
Cochez cette case afin de récupérer les attributs personnalisés avec chaque message.

**Custom visibility timeout**
Cochez cette case et, dans le champ qui s’affiche, spécifiez le délai avant suspension de la visibilité (en secondes) pour les messages reçus. Si aucun délai n’est spécifié, le délai avant suspension de la visibilité globale des la file sera utilisé pour les messages reçus. Pour plus d’informations, consultez Délai de visibilité.

**Custom wait time**
Cochez cette case et, dans le champ qui s’affiche, spécifiez la durée (en secondes) durant laquelle l’appel attend que le message arrive dans la file avant que l’appel soit retourné.

**Delete the messages while streaming**
Cochez cette case pour supprimer le message en le récupérant de la file.

**Read all messages from the queue**
Cochez cette case pour récupérer tous les messages de la file.
Cette case n’est pas disponible lorsque la case **Delete the messages while streaming** est décochée.

**Max number of message to return per request**
Spécifiez le nombre maximum de messages à retrouver pour chaque requête. Les valeurs valides sont comprises entre 1 et 10.

**Die on error**
Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.
Décrochez la case pour ignorer les lignes en erreur et terminer le processus.

### Advanced settings

**Config client**
Cochez cette case et, dans la table affichée, spécifiez les paramètres de configuration de la table. Cliquez sur le bouton [+] sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre de configuration client et configurez la valeur des attributs suivants pour chaque paramètre :
- **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.
- **Value** : saisissez la valeur du paramètre correspondant.

**STS Endpoint**
Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token.
Récupérer des messages d'une file Amazon SQS

Voici un exemple d'utilisation des composants Talend pour récupérer tous les messages d’une file Amazon SQS (Simple Queue Service) existante et supprimer la file vide.

Créer un Job pour récupérer des messages Amazon SQS

Créez un Job pour ouvrir une connexion afin d'accéder à Amazon SQS, récupérer tous les messages d’une file Amazon SQS existante et supprimer la file.

```plaintext
#Creator;Body
Talend DOC Team;the 1st message from Talend DOC team
Talend DEV Team;the 2nd message from Talend DEV team
Talend QA Team;the 3rd message from Talend QA team
```

**Procédure**

1. Créez un nouveau Job et ajoutez un composant `tSQSConnection`, un `tSQSInput`, un `tLogRow` et un `tSQSQueueDelete` en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le `tSQSInput` au `tLogRow` à l’aide d’un lien `Row > Main`.
3. Reliez le `tSQSConnection` au `tSQSInput` à l’aide d’un lien `Trigger > OnSubjobOk`.
4. Reliez le `tSQSInput` au `tSQSQueueDelete` à l’aide d’un lien `Trigger > OnSubjobOk`.

**Ouvrir une connexion pour accéder à Amazon SQS**

Configurez le `tSQSConnection` pour ouvrir une connexion à Amazon SQS et y accéder.

**Procédure**

1. Double-cliquez sur le `tSQSConnection` pour ouvrir sa vue `Basic settings`. 
2. Dans les champs **Access Key** et **Secret Key**, spécifiez les informations d’authentification requises pour accéder à Amazon SQS.

3. Sélectionnez une région AWS disponible pour Amazon SQS, dans la liste **Region**. Dans cet exemple, sélectionnez **Asia Pacific (Tokyo)**.

### Récupérer des messages d’une file Amazon SQS

Configurez le composant **tSQSInput** et le **tLogRow** pour récupérer tous les messages d’une file Amazon SQS existante et afficher les messages récupérés dans la console du Studio Talend.

#### Procédure

1. Double-cliquez sur le composant **tSQSInput** pour ouvrir sa vue **Basic settings**.

2. Spécifiez les informations de connexion requises pour accéder à Amazon SQS. Dans cet exemple, cochez la case **Use an existing connection** et, dans la liste **Component List** qui s’affiche, sélectionnez le composant de connexion afin de réutiliser ses informations de connexion précédemment saisies.

3. Dans le champ **Queue (Name or URL)**, saisissez le nom de la file de laquelle les messages seront récupérés. Dans cet exemple, saisissez **talend**.

Vous pouvez voir que le schéma du tSQSInput est prédéfini et contient quatre colonnes de type String : MessageId, ReceiptHandle, MD5OfBody et Body.

5. Cliquez cinq fois sur le bouton + pour ajouter cinq colonnes de type String, dont quatre pour des attributs Amazon SQS ApproximateFirstReceiveTimestamp, ApproximateReceiveCount, SenderId et SentTimestamp et une pour l'attribut personnalisé Creator.

6. Cliquez sur OK afin de sauvegarder les modifications et fermer la fenêtre du schéma.

7. Cochez les cases Read standard attributes et Read custom user attributes afin de récupérer la valeur des quatre attributs Amazon SQS et de l'attribut personnalisé, avec chaque message.

8. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings, puis sélectionnez l'option Table (print values in cells of a table) dans la zone Mode pour une meilleure lisibilité des résultats.

Supprimer une file Amazon SQS

Configurez le composant tSQSQueueDelete pour créer une file Amazon SQS.

Procédure

1. Double-cliquez sur le tSQSQueueDelete pour ouvrir sa vue Basic settings.
2. Spécifiez les détails de connexion requis pour accéder à Amazon SQS. Dans cet exemple, cochez la case *Use an existing connection* et, dans la liste *Component List* qui s'affiche, sélectionnez le composant de connexion, afin de réutiliser les informations précédemment définis.

3. Dans le champ *Queue (Name or URL)*, saisissez le nom de la file à supprimer. Dans cet exemple, saisissez `talend`.

**Exécuter le Job pour récupérer des messages Amazon SQS**

Une fois le Job et les composants configurés pour récupérer des messages Amazon SQS, vous pouvez exécuter le Job et vérifier ses résultats d'exécution.

**Procédure**

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder le Job, puis sur **F6** pour l'exécuter.

2. Comme affiché dans l'image ci-dessus, les trois messages et leurs attributs sont récupérés et affichés dans la console.

3. Regardez les détails de la file dans la console AWS afin de valider les résultats d'exécution du Job.
Vous pouvez voir que la file talend a été supprimée et n’est plus dans la console AWS.
**tSQSMessageChangeVisibility**

Ce composant modifie le temps durant lequel un message spécifié est visible dans une file d’attente Amazon SQS (Simple Queue Service).

### Propriétés du tSQSMessageChangeVisibility Standard

Ces propriétés sont utilisées pour configurer le tSQSMessageChangeVisibility s’exécutant dans un framework de Jobs Standard.

Le composant tSQSMessageChangeVisibility Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Basic setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Access Key et Secret Key</td>
<td>Spécifiez les clés d’accès (l’ID de la clé d’accès dans le champ Access Key et la clé secrète d’accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés d’accès d’AWS, consultez Clés d’accès (ID de clé d’accès et clé d’accès secrète).</td>
</tr>
<tr>
<td></td>
<td>Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
<tr>
<td>Assume Role</td>
<td>Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.</td>
</tr>
<tr>
<td></td>
<td>• Role ARN : nom Amazon Resource Name (ARN) du rôle.</td>
</tr>
<tr>
<td></td>
<td>• Role session name : identifiant de la session du rôle.</td>
</tr>
<tr>
<td></td>
<td>• Session duration (minutes) : durée (en minutes) pour laquelle est active la session du rôle.</td>
</tr>
</tbody>
</table>
Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).

**Region**
Spécifiez la région AWS en sélectionnant un nom de région ou en saisissant une région entre guillemets doubles (par exemple ’us-east-1’) dans la liste. Pour plus d’informations concernant les régions AWS, consultez Régions et points de terminaison AWS.

**Queue (Name or URL)**
Spécifiez le nom ou l’URL de la file dans laquelle se trouve le message duquel changer la période de visibilité.

**Receipt Handle**
Spécifiez le mode d’accusé réception associé au message dont la période de visibilité va être modifiée.

**Visibility Timeout in Seconds**
Saisissez la nouvelle valeur pour la période de visibilité, en secondes, pour le message spécifié. Les valeurs valides sont comprises entre 0 et 43200.

**Die on error**
Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.
Décochez la case pour ignorer les lignes en erreur et terminer le processus.

### Advanced settings

**Config client**
Cochez cette case et, dans la table affichée, spécifiez les paramètres de configuration de la Table. Cliquez sur le bouton [+] sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre de configuration client et configurez la valeur des attributs suivants pour chaque paramètre :
- **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.
- **Value** : saisissez la valeur du paramètre correspondant.

**STS Endpoint**
Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.
Cette case est disponible uniquement lorsque la case **Assume role** est cochée.

**tStatCatcher Statistics**
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global variables

**ERROR_MESSAGE**
Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable
fonctionne uniquement lorsque la case **Die on error** est cochée.

### Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job. |

### Scénario associé

Aucun scénario n’est disponible pour ce composant.
### tSQSMessageDelete

Ce composant supprime un message spécifié d’une file Amazon SQS (Simple Queue Service).

### Propriétés du tSQSMessageDelete Standard

Ces propriétés sont utilisées pour configurer le `tSQSMessageDelete` s’exécutant dans un framework de Jobs Standard.

Le composant tSQSMessageDelete Standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

#### Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Access Key et Secret Key**

Spécifiez les clés d’accès (l’ID de la clé d’accès dans le champ Access Key et la clé secrète d’accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés d’accès d’AWS, consultez Clés d’accès (ID de clé d’accès et clé d’accès secrète).

Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.

**Inherit credentials from AWS role**

Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.

**Assume Role**

Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.

- **Role ARN** : nom Amazon Resource Name (ARN) du rôle.
- **Role session name** : identifiant de la session du rôle.
### Session duration (minutes)
- Durée (en minutes) pour laquelle est active la session du rôle.

Pour plus d'informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).

### Region
- Spécifiez la région AWS en sélectionnant un nom de région ou en saisisant une région entre guillemets doubles (par exemple "us-east-1") dans la liste. Pour plus d'informations concernant les régions AWS, consultez Régions et points de terminaison AWS.

### Queue (Name or URL)
- Spécifiez le nom ou l'URL de la file de laquelle le message sera supprimé.

### Receipt Handle
- Spécifiez le mode d'accusé réception associé au message à supprimer.

### Die on error
- Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.
- Décrochez la case pour ignorer les lignes en erreur et terminer le processus.

### Advanced settings

#### Config client
- Cochez cette case et, dans la table affichée, spécifiez les paramètres de configuration de la table. Cliquez sur le bouton [+] sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre de configuration client et configurez la valeur des attributs suivants pour chaque paramètre :
  - **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.
  - **Value** : saisissez la valeur du paramètre correspondant.

#### STS Endpoint
- Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.
- Cette case est disponible uniquement lorsque la case Assume role est cochée.

#### tStatCatcher Statistics
- Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global variables

#### ERROR_MESSAGE
- Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case Die on error est cochée.
Utilisation

| Règle d'utilisation | Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job. |

Scénario associé

Aucun scénario n'est disponible pour ce composant.
**tSQSOutput**

Ce composant livre un ou plusieurs messages dans une file Amazon SQS (Simple Queue Service).

**Propriétés du tSQSOutput standard**

Ces propriétés sont utilisées pour configurer le tSQSOutput s'exécutant dans le framework des Jobs standard.

Le composant tSQSOutput standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Key et Secret Key</td>
<td>Spécifiez les clés d'accès (l'ID de la clé d'accès dans le champ Access Key et la clé secrète d'accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d'informations concernant les clés d'accès d'AWS, consultez Clés d'accès (ID de clé d'accès et clé d'accès secrète). Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour tirer parti des informations d'authentification du profil de l'instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d'Amazon EC2. Pour utiliser cette option, votre Job doit s'exécuter dans Amazon EC2 ou d'autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d'informations, consultez Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2.</td>
</tr>
</tbody>
</table>
| Assume Role | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  
- **Role ARN** : nom Amazon Resource Name (ARN) du rôle.  
- **Role session name** : identifiant de la session du rôle.  
- **Session duration (minutes)** : durée (en minutes) pour laquelle est active la session du rôle.  
Pour plus d'informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais). |
<table>
<thead>
<tr>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spécifiez la région AWS en sélectionnant un nom de région ou en saisissant une région entre guillemets doubles (par exemple ‘us-east-1’) dans la liste. Pour plus d’informations concernant les régions AWS, consultez Régions et points de terminaison AWS.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Queue (Name or URL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spécifiez le nom ou l’URL de la file dans laquelle livrer les messages.</td>
</tr>
<tr>
<td>Le nom d’une file FIFO doit se terminer par le suffixe .fifo.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Schema et Edit schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td>Le schéma de ce composant est prédéfini et contient les colonnes suivantes. Vous pouvez cliquer sur le bouton [...] à côté du champ Edit schema pour voir son schéma et le modifier, en ajoutant des attributs personnalisés pour les messages à livrer.</td>
</tr>
<tr>
<td>• <strong>Body</strong> : corps du message à livrer.</td>
</tr>
<tr>
<td>• <strong>MessageGroupId</strong> : groupe spécifique de messages auquel le message appartient. Disponible uniquement lorsque la case Use FIFO queues (first-in-first-out) est cochée.</td>
</tr>
<tr>
<td>• <strong>MessageDeduplicationId</strong> : jeton utilisé pour le dédoublonnage du message. Disponible uniquement lorsque la case Use FIFO queues (first-in-first-out) est cochée et que la case Use Content Based Deduplication est décochée.</td>
</tr>
<tr>
<td>Pour plus d’informations concernant les colonnes prédéfinies, consultez SendMessage (en anglais).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use batch mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour livrer les messages dans la file à l’aide d’une requête par lots. Dans le champ Batch size qui s’affiche, saisissez le nombre de messages à livrer par lot. Notez que le nombre maximal de messages à livrer par lot est de 10.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use FIFO (first-in-first-out) queues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour livrer des messages dans une file FIFO.</td>
</tr>
<tr>
<td>Pour plus d’informations sur les files FIFO, consultez Logique de la file d’attente FIFO.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Content Based Deduplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour activer le dédoublonnage du contenu. Lorsque cette case est cochée, Amazon SQS utilise un hachage SHA-256 pour générer l’ID de dédoublonnage du message, à l’aide du corps du message.</td>
</tr>
<tr>
<td>Cette propriété est disponible uniquement lorsque la case Use FIFO (first-in-first-out) queues est cochée.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Die on error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.</td>
</tr>
</tbody>
</table>
Décochez la case pour ignorer les lignes en erreur et terminer le processus.

### Advanced settings

#### Config client

Cochez cette case et, dans la table affichée, spécifiez les paramètres de configuration de la table. Cliquez sur le bouton [+] sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre de configuration client et configurez la valeur des attributs suivants pour chaque paramètre :

- **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.
- **Value** : saisissez la valeur du paramètre correspondant.

#### STS Endpoint

Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.

Cette case est disponible uniquement lorsque la case **Assume role** est cochée.

#### tStatCatcher Statistics

Cochez cette case pour collecter les métdonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Variables globales

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_SUCCESS</td>
<td>Nombre de lignes traitées avec succès. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_REJECTED</td>
<td>Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case <strong>Die on error</strong> est cochée.</td>
</tr>
</tbody>
</table>

### Utilisation

Règle d’utilisation

Ce composant est généralement utilisé comme composant de fin dans un Job ou un sous-job et nécessite un lien d’entrée.
Livrer des messages dans une file Amazon SQS

Voici un exemple d’utilisation des composants Talend pour créer une file Amazon SQS (Simple Queue Service), livrer quelques messages dans la file, puis obtenir et afficher les attributs de la file dans la console.

Les messages d'entrée dans cet exemple se présentent comme suit et chaque message contient l'attribut de l'auteur du message, ainsi que les informations du corps du message.

#Creator;Body
Talend DOC Team;the 1st message from Talend DOC team
Talend DEV Team;the 2nd message from Talend DEV team
Talend QA Team;the 3rd message from Talend QA team

Créer un Job pour livrer des messages Amazon SQS

Créez un Job pour ouvrir une connexion afin d'accéder à Amazon SQS, créez une file Amazon SQS, livrez des messages à la file, puis récupérez et affichez les attributs de la file dans la console.

Procédure

1. Créez un nouveau Job et ajoutez un tSQSConnection, un tSQSQueueCreate, un tFixedFlowInput, un tSQSOutput, un tSQSQueueAttributes et un tLogRow en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le composant tFixedFlowInput au tSQSOutput à l'aide d’un lien Row > Main.
3. Répétez cette étape pour relier le tSQSQueueAttributes au tLogRow.
4. Répétez l'opération pour relier le tSQSConnection au tSQSQueueCreate à l'aide d’un lien Trigger > OnSubjobOk.
5. Répétez l’opération pour relier le composant tSQSQueueCreate au tFixedFlowInput et le tFixedFlowInput au tSQSQueueAttributes.
Ouvrir une connexion pour accéder à Amazon SQS

Configurez le tSQSConnection pour ouvrir une connexion et accéder à Amazon SQS.

**Procédure**

1. Double-cliquez sur le tSQSConnection pour ouvrir sa vue Basic settings.

2. Dans les champs **Access Key** et **Secret Key**, spécifiez les informations d’authentification requises pour accéder à Amazon SQS.

3. Sélectionnez une région AWS disponible pour Amazon SQS dans la liste **Region**. Dans cet exemple, sélectionnez **Asia Pacific (Tokyo)**.

**Créer une file Amazon SQS**

Configurez le composant tSQSQueueCreate afin de créer une file Amazon SQS.

**Procédure**

1. Double-cliquez sur le tSQSQueueCreate pour ouvrir sa vue Basic settings.

2. Spécifiez les informations de connexion requises pour accéder à Amazon SQS. Dans cet exemple, cochez la case **Use an existing connection** et, dans la liste Component List qui s’affiche, sélectionnez le composant de connexion afin de réutiliser les informations précédemment définies.

3. Dans le champ **Queue (Name or URL)**, spécifiez le nom de la file à créer. Dans cet exemple, saisissez **talend**.

4. Si nécessaire, spécifiez les attributs de la file à créer, dans la table **Queue Attributes**. Dans cet exemple, l’attribut **MessageRetentionPeriod** est ajouté et sa valeur est définie à 86400, ce qui signifie qu’Amazon SQS va retenir les messages livrés dans la file durant 86 400 secondes (une journée) au lieu de la valeur par défaut de 345 600 secondes (quatre jours).
**Livrer des messages dans la file Amazon SQS**

Configurez les composants `tFixedFlowInput` et `tSQSOutput` pour livrer des messages à la nouvelle file vide Amazon SQS.

**Procédure**

1. Double-cliquez sur le composant `tSQSOutput` pour ouvrir sa vue **Basic settings**.

2. Spécifiez les informations de connexion requises pour accéder à Amazon SQS. Dans cet exemple, cochez la case **Use an existing connection** et, dans la liste **Component List** qui s’affiche, sélectionnez le composant de connexion afin de réutiliser les informations de connexion précédemment définies.

3. Dans le champ **Queue (Name or URL)**, saisissez le nom de la file dans laquelle les messages seront livrés. Dans cet exemple, saisissez `talend`.


   Vous pouvez constater que le schéma du `tSQSOutput` est prédéfini et contient une seule colonne de type String, nommée **Body**, stockant le corps de chaque message à livrer.

5. Cliquez sur le bouton dans le panneau de droite afin d’ajouter une colonne, nommée **Creator**, de type **String**, qui contiendra la valeur de l’attribut personnalisé de l’auteur de chaque message.

6. Cliquez sur le bouton afin de copier toutes les colonnes du schéma de sortie dans le schéma d’entrée. Dans la boîte de dialogue qui s’ouvre, cliquez sur **OK** pour accepter les modifications.

7. Cliquez sur **OK** pour fermer le schéma.

8. Double-cliquez sur le `tFixedFlowInput` pour ouvrir sa vue **Basic settings**.
Dans la zone **Mode**, sélectionnez **Use Inline Content(delimited file)**. Dans le champ **Content**, saisissez l’attribut de l’auteur et les données du corps des messages à livrer dans Amazon SQS. Dans cet exemple, saisissez :

```
Talend DOC Team;the 1st message from Talend DOC team
Talend DEV Team;the 2nd message from Talend DEV team
Talend QA Team;the 3rd message from Talend QA team
```

**Récupération d’attributs de la file Amazon SQS**

Configurez les composants **tSQSQueueAttributes** et **tLogRow** pour récupérer et afficher les attributs de la file Amazon SQS dans la console.

**Procédure**

1. Double-cliquez sur le composant **tSQSQueueAttributes** pour ouvrir sa vue **Basic settings**.
2. Spécifiez les détails de connexion nécessaires pour accéder à Amazon SQS. Dans cet exemple, cochez la case **Use an existing connection**, puis, dans la liste déroulante **Component List** qui s’affiche, sélectionnez le composant de connexion dont vous souhaitez réutiliser les détails de connexion.
3. Dans le champ **Queue (Name or URL)**, indiquez le nom de la file dont les attributs seront récupérés. Dans cet exemple, il s’agit de **talend**.
4. Double-cliquez sur le composant **tLogRow** afin d’ouvrir sa vue **Basic settings**.
5. Dans la zone **Mode**, sélectionnez **Vertical** afin d’afficher chaque attribut et sa valeur sur une même ligne.

**Exécuter le Job pour livrer des messages dans Amazon SQS**

Après avoir configuré le Job et ses composants, afin de livrer des messages Amazon SQS, vous pouvez exécuter le Job et vérifier ses résultats d’exécution.

**Procédure**

1. Appuyez sur les touches **Ctrl + S** afin de sauvegarder le Job, puis sur **F6** pour l’exécuter.
Comme affiché ci-dessus, le Job a bien été exécuté et les attributs de la file sont affichés dans la console.

2. Regardez les détails de la file dans la console AWS, afin de valider les résultats d’exécution du Job.

Vous pouvez voir que la file talend a bien été créée, avec la période de rétention d’une journée. Trois messages sont disponibles dans la file.

Vous pouvez constater que les trois messages sont disponibles dans la file.


Vous pouvez voir que la valeur de l’attribut **Creator** personnalisé pour le message s’affiche dans l’onglet **Message Attributes**.
tSQSQueueAttributes

Ce composant récupère les attributs d’une file Amazon SQS (Simple Queue Service) spécifiée.

**Propriétés du tSQSQueueAttributes Standard**

Ces propriétés sont utilisées pour configurer le tSQSQueueAttributes s’exécutant dans le framework de Jobs Standard.

Le composant tSQSQueueAttributes standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>
| Access Key et Secret Key         | Spécifiez les clés d’accès (l'ID de la clé d’accès dans le champ **Access Key** et la clé secrète d’accès dans le champ **Secret Key**) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés d’accès d’AWS, consultez Clés d’accès (ID de clé d’accès et clé d’accès secrète).

Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ **Secret Key**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. |
| Inherit credentials from AWS role| Cochez cette case pour tirer parti des informations d’autentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2. |
| Assume Role                      | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.

- **Role ARN** : nom Amazon Resource Name (ARN) du rôle.
- **Role session name** : identifiant de la session du rôle.
- **Session duration (minutes) : durée (en minutes) pour laquelle est active la session du rôle.** |
Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).

**Region**
Spécifiez la région AWS en sélectionnant un nom de région ou en saisissant une région entre guillemets doubles (par exemple "us-east-1") dans la liste. Pour plus d’informations concernant les régions AWS, consultez Régions et points de terminaison AWS.

**Queue (Name or URL)**
Spécifiez le nom ou l’URL de la file dont proviennent les informations d’attributs.

**Schema et Edit schema**
Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

Le schéma de ce composant est en lecture seule et contient deux colonnes prédéfinies dans lesquelles sont décrits les attributs de la file spécifiée. Vous pouvez cliquer sur le bouton «...» à côté de `Edit schema` pour en afficher le schéma. Pour en savoir plus sur chaque attribut, consultez la page GetQueueAttributes (en anglais).

**Die on error**
Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

Décrochez la case pour ignorer les lignes en erreur et terminer le processus.

### Advanced settings

**Config client**
Cochez cette case et, dans la table affichée, spécifiez les paramètres de configuration de la table. Cliquez sur le bouton [+ ] sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre de configuration client et configurez la valeur des attributs suivants pour chaque paramètre :

- **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.
- **Value** : saisissez la valeur du paramètre correspondant.

**STS Endpoint**
Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.

Cette case est disponible uniquement lorsque la case **Assume role** est cochée.

**tStatCatcher Statistics**
Cochez cette case pour collecter les métdonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.
Variables globales

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case Die on error est cochée. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie. |

Scénario associé

Livrer des messages dans une file Amazon SQS à la page 3980
tSQSQueueCreate

Ce composant crée une nouvelle file Amazon SQS (Simple Queue Service).

**Propriétés du tSQSQueueCreate standard**

Ces propriétés sont utilisées pour configurer le tSQSQueueCreate s’exécutant dans le framework des Jobs standard.

Le composant tSQSQueueCreate standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th><strong>Use an existing connection</strong></th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
</table>
| **Access Key et Secret Key**  | Spécifiez les clés d’accès (l’ID de la clé d’accès dans le champ Access Key et la clé secrète d’accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés d’accès d’AWS, consultez Clés d’accès (ID de clé d’accès et clé d’accès secrète).  
  Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres. |
| **Inherit credentials from AWS role** | Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2. |
| **Assume Role** | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  
  - Role ARN : nom Amazon Resource Name (ARN) du rôle.  
  - Role session name : identifiant de la session du rôle.  
  - Session duration (minutes) : durée (en minutes) pour laquelle est active la session du rôle.  
  Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais). |
<table>
<thead>
<tr>
<th>Region</th>
<th>Spécifiez la région AWS en sélectionnant un nom de région ou en saisissant une région entre guillemets doubles (par exemple 'us-east-1') dans la liste. Pour plus d'informations concernant les régions AWS, consultez Régions et points de terminaison AWS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue (Name or URL)</td>
<td>Spécifiez le nom de la file à créer.</td>
</tr>
</tbody>
</table>
| Queue Attributes | Spécifiez les attributs de la file à créer. Cliquez sur le bouton + sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un attribut de la file et configurez la valeur des paramètres suivants pour chaque attribut.  
- **Attribute** : cliquez dans la cellule et sélectionnez le nom de l'attribut dans la liste déroulante.  
- **Value** : saisissez la valeur de l'attribut correspondant.  
Si aucune valeur n'est définie pour les attributs, la file aura la valeur par défaut pour cet attribut.  
Pour plus d'informations concernant les attributs de la file, consultez CreateQueue > Request Parameters (en anglais). |
| Die on error | Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.  
Décocochez la case pour ignorer les lignes en erreur et terminer le processus. |

**Advanced settings**

| Config client | Cochez cette case et, dans la table affichée, spécifiez les paramètres de configuration de la table. Cliquez sur le bouton + sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre de configuration client et configurez la valeur des attributs suivants pour chaque paramètre :  
- **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.  
- **Value** : saisissez la valeur du paramètre correspondant. |
| STS Endpoint | Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.  
Cette case est disponible uniquement lorsque la case **Assume role** est cochée. |
| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |
Variables globales

| **ERROR_MESSAGE** | Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case **Die on error** est cochée. |

Utilisation

| Règle d’utilisation | Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job. |

Scénario associé

Livrer des messages dans une file Amazon SQS à la page 3980
tSQSQueueDelete

Ce composant supprime une file Amazon SQS (Simple Queue Service).

**Propriétés du tSQSQueueDelete standard**

Ces propriétés sont utilisées pour configurer le tSQSQueueDelete s’exécutant dans le framework des Jobs standard.

Le composant tSQSQueueDelete standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td>Access Key et Secret Key</td>
<td>Spécifiez les clés d’accès (l’ID de la clé d’accès dans le champ Access Key et la clé secrète d’accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés d’accès d’AWS, consultez Clés d’accès (ID de clé d’accès et clé d’accès secrète). Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
<tr>
<td>Assume Role</td>
<td>Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.</td>
</tr>
<tr>
<td></td>
<td>• Role ARN : nom Amazon Resource Name (ARN) du rôle.</td>
</tr>
<tr>
<td></td>
<td>• Role session name : identifiant de la session du rôle.</td>
</tr>
<tr>
<td></td>
<td>• Session duration (minutes) : durée (en minutes) pour laquelle est active la session du rôle.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).</td>
</tr>
</tbody>
</table>
Region

Spécifiez la région AWS en sélectionnant un nom de région ou en saisissant une région entre guillemets doubles (par exemple "us-east-1") dans la liste. Pour plus d’informations concernant les régions AWS, consultez Régions et points de terminaison AWS.

Queue (Name or URL)

Spécifiez le nom ou l’URL de la file à supprimer.

Die on error

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

Décochez la case pour ignorer les lignes en erreur et terminer le processus.

Advanced settings

Config client

Cochez cette case et, dans la table affichée, spécifiez les paramètres de configuration de la table. Cliquez sur le bouton [+] sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre de configuration client et configurez la valeur des attributs suivants pour chaque paramètre :

- **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.
- **Value** : saisissez la valeur du paramètre correspondant.

STS Endpoint

Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.

Cette case est disponible uniquement lorsque la case **Assume role** est cochée.

tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

Variables globales

ERROR_MESSAGE

Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case **Die on error** est cochée.

Utilisation

Règle d’utilisation

Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job.
Scénario associé

Récupérer des messages d'une file Amazon SQS à la page 3965
tSQSQueueList

Ce composant effectue des itérations et liste l’URL des files Amazon SQS (Simple Queue Service) dans une région spécifiée.

Propriétés du tSQSQueueList standard

Ces propriétés sont utilisées pour configurer le tSQSQueueList s’exécutant dans le framework des Jobs standard.
Le composant tSQSQueueList appartient à la famille Cloud.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Key et Secret Key</td>
<td>Spécifiez les clés d’accès (‘ID de la clé d’accès dans le champ Access Key et la clé secrète d’accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés d’accès d’AWS, consultez Clés d’accès (ID de clé d’accès et clé d’accès secrète). Pour saisir la clé secrète, cliquez sur le bouton […] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
</tbody>
</table>
| Assume Role | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle.  
  • Role ARN : nom Amazon Resource Name (ARN) du rôle.  
  • Role session name : identifiant de la session du rôle.  
  • Session duration (minutes) : durée (en minutes) pour laquelle est active la session du rôle. |
<table>
<thead>
<tr>
<th>tSQSQueueList</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spécifiez la région AWS en sélectionnant un nom de région ou en saisissant une région entre guillemets doubles (par exemple 'us-east-1') dans la liste. Pour plus d’informations concernant les régions AWS, consultez Régions et points de terminaison AWS.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>Config client</th>
</tr>
</thead>
</table>
| Cochez cette case et, dans la table affichée, spécifiez les paramètres de configuration de la table. Cliquez sur le bouton + sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre de configuration client et configurez la valeur des attributs suivants pour chaque paramètre :  
| - Client Parameter : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.  
| - Value : saisissez la valeur du paramètre correspondant. |

<table>
<thead>
<tr>
<th>STS Endpoint</th>
</tr>
</thead>
</table>
| Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées.  
Cette case est disponible uniquement lorsque la case Assume role est coché. |

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Variables globales

<table>
<thead>
<tr>
<th>CURRENT_QUEUE_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL de la file en cours de traitement. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NB_QUEUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nombre de files traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ERROR_MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ce composant est généralement utilisé en tant que composant de début dans un Job ou un sous-job et nécessite un lien de sortie.</td>
</tr>
</tbody>
</table>
Lister des files Amazon SQS dans une région AWS

Voici un exemple d’utilisation des composants Talend pour itérer et lister les URL de toutes les files Amazon SQS (Simple Queue Service) dans une région AWS.

Créer un Job pour lister les files Amazon SQS

Créez un Job pour ouvrir une connexion pour accéder à Amazon SQS, puis effectuer une itération et lister toutes les URL des files Amazon SQS dans une région spécifiée, pour ensuite afficher le nombre total de files dans la région.

Prérequis : vous devez créer plusieurs files Amazon SQS dans une région AWS afin de reproduire ce scénario. Dans cet exemple, quatre files talend, talend-dev, talend-doc et talend-qa ont été créées dans la région Asia Pacific (Tokyo), comme l’indique la capture d’écran ci-dessous.

Procédure

1. Créez un nouveau Job et ajoutez un composant tSQSConnection, un tSQSQueueList, un tIterateToFlow, un tLogRow et un tJava, en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le `tSQSQueueList` au `tIterateToFlow` à l’aide d’un lien `Row > Iterate`.
3. Reliez le `tIterateToFlow` au `tLogRow` à l’aide d’un lien `Row > Main`.
4. Reliez le composant `tSQSConnection` au `tSQSQueueList` à l’aide d’un lien `Trigger > OnSubjobOk`.
5. Reliez le `tSQSQueueList` au `tJava` à l’aide d’un lien `Trigger > OnSubjobOk`.

**Ouvrir une connexion pour accéder à Amazon SQS**

Configurez le `tSQSConnection` pour ouvrir une connexion à Amazon SQS et y accéder.

**Procédure**

1. Double-cliquez sur le `tSQSConnection` pour ouvrir sa vue `Basic settings`.

   ![tSQSConnection](image)

2. Dans les champs `Access Key` et `Secret Key`, spécifiez les informations d’authentification requises pour accéder à Amazon SQS.
3. Sélectionnez une région AWS disponible pour Amazon SQS dans la liste `Region`. Dans cet exemple, sélectionnez `Asia Pacific (Tokyo)`.

**Lister toutes les files Amazon SQS dans une région AWS**

Configurez le composant `tSQSQueueList`, le `tIterateToFlow`, le `tLogRow` et le `tJava` pour lister toutes les URL des files Amazon SQS dans une région AWS et afficher le nombre total de files dans la console du Studio Talend.

**Procédure**

1. Double-cliquez sur le composant `tSQSQueueList` pour ouvrir sa vue `Basic settings`.

   ![tSQSQueueList](image)

2. Spécifiez les informations de connexion requises pour accéder à Amazon SQS. Dans cet exemple, cochez la case `Use an existing connection` et, dans la liste `Component List` qui s’affiche, sélectionnez le composant de connexion afin de réutiliser les informations précédemment définies.
3. Double-cliquez sur le `tIterateToFlow` pour ouvrir sa vue `Basic settings`.
4. Cliquez sur le bouton à côté du champ Edit schema pour ouvrir le schéma.

5. Cliquez sur le bouton pour ajouter une colonne de type String nommée CurrentQueueURL qui contiendra les URLs des files à lister. Cela fait, cliquez sur OK pour fermer le schéma.

6. Dans la table Mapping, configurez la valeur de la colonne CurrentQueueURL. Dans cet exemple, la valeur est configurée à ((String)globalMap.get("tSQSQueueList_1_CURRENT_QUEUE_NAME")), qui est la valeur de la variable globale CURRENT_QUEUE_NAME pour le composant tSQSQueueList. Notez que vous pouvez renseigner la valeur en appuyant sur les touches Ctrl + Espace pour accéder à la liste des variables globales et sélectionner tSQSQueueList_1_CURRENT_QUEUE_NAME dans la liste.

7. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings et sélectionnez Table (print values in cells of a table) dans la zone Mode, pour une meilleure lisibilité des résultats.

8. Double-cliquez sur le composant tJava pour ouvrir sa vue Basic settings.
9. Dans le champ **Code**, saisissez le code suivant pour afficher le nombre total de files dans la région.

```java
System.out.println("The number of queues: "+((Integer)globalMap.get("tSQ
SQueueList_1_NB_QUEUE")));```

**Exécuter le Job pour lister les files Amazon SQS**

Après avoir configuré le Job et ses composants pour lister les files Amazon SQS, vous pouvez exécuter le Job et vérifier les résultats d'exécution du Job.

**Procédure**

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour l'exécuter.

```
[statistics] connecting to socket on port 3699
[statistics] connected
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CurrentQueueURL</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td><a href="https://sqs.ap-northeast-1.amazonaws.com/855211627039/talend">https://sqs.ap-northeast-1.amazonaws.com/855211627039/talend</a></td>
<td></td>
</tr>
<tr>
<td><a href="https://sqs.ap-northeast-1.amazonaws.com/855211627039/talend-dev">https://sqs.ap-northeast-1.amazonaws.com/855211627039/talend-dev</a></td>
<td></td>
</tr>
<tr>
<td><a href="https://sqs.ap-northeast-1.amazonaws.com/855211627039/talend-doc">https://sqs.ap-northeast-1.amazonaws.com/855211627039/talend-doc</a></td>
<td></td>
</tr>
<tr>
<td><a href="https://sqs.ap-northeast-1.amazonaws.com/855211627039/talend-qa">https://sqs.ap-northeast-1.amazonaws.com/855211627039/talend-qa</a></td>
<td></td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>The number of queues: 4</td>
<td>(Tokyo) et le nombre total de files sont affichés dans la console.</td>
</tr>
</tbody>
</table>
```

Comme affiché ci-dessus, les URL de toutes les files de la région **Asia Pacific (Tokyo)** et le nombre total de files sont affichés dans la console.
tSQSQueuePurge

Ce composant purge les messages dans une file Amazon SQS (Simple Queue Service).

Propriétés du tSQSQueuePurge standard

Ces propriétés sont utilisées pour configurer le tSQSQueuePurge s’exécutant dans le framework des Jobs standard.

Le composant tSQSQueuePurge standard appartient à la famille Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Key et Secret Key</td>
<td>Spécifiez les clés d’accès (l’ID de la clé d’accès dans le champ Access Key et la clé secrète d’accès dans le champ Secret Key) requises pour accéder à Amazon Web Services. Pour plus d’informations concernant les clés d’accès d’AWS, consultez Clés d’accès (ID de clé d’accès et clé d’accès secrète). Pour saisir la clé secrète, cliquez sur le bouton [...] à côté du champ Secret key, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles puis cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Inherit credentials from AWS role</td>
<td>Cochez cette case pour tirer parti des informations d’authentification du profil de l’instance. Ces informations peuvent être utilisée sur des instances Amazon EC2 et sont fournies via le service de métadonnées d’Amazon EC2. Pour utiliser cette option, votre Job doit s’exécuter dans Amazon EC2 ou d’autres services pouvant tirer parti des rôles IAM pour accéder aux ressources. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2.</td>
</tr>
</tbody>
</table>
| Assume Role               | Cochez cette case et spécifiez les valeurs des paramètres utilisés pour créer une nouvelle session du rôle. 

- **Role ARN** : nom Amazon Resource Name (ARN) du rôle.
- **Role session name** : identifiant de la session du rôle.
- **Session duration (minutes)** : durée (en minutes) pour laquelle est active la session du rôle.  
Pour plus d’informations concernant les rôles et AssumeRole, consultez AssumeRole (en anglais). |
<table>
<thead>
<tr>
<th>Region</th>
<th>Spécifiez la région AWS en sélectionnant un nom de région ou en saisissant une région entre guillemets doubles (par exemple ’us-east-1’) dans la liste. Pour plus d’informations concernant les régions AWS, consultez Régions et points de terminaison AWS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue (Name or URL)</td>
<td>Spécifiez le nom ou l’URL de la file de laquelle purger les messages.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient. Décochez la case pour ignorer les lignes en erreur et terminer le processus.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| Config client | Cochez cette case et, dans la table affichée, spécifiez les paramètres de configuration de la table. Cliquez sur le bouton [+] sous la table pour ajouter autant de lignes que nécessaire, chaque ligne pour un paramètre de configuration client et configurez la valeur des attributs suivants pour chaque paramètre :

  - **Client Parameter** : cliquez dans la cellule et sélectionnez un paramètre dans la liste déroulante.
  - **Value** : saisissez la valeur du paramètre correspondant. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>STS Endpoint</td>
<td>Cochez cette case et, dans le champ qui s’affiche, spécifiez l’endpoint du service AWS Security Token Service duquel les informations d’authentification sont récupérées. Cette case est disponible uniquement lorsque la case Assume role est cochée.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Variables globales**

| ERROR_MESSAGE | Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement lorsque la case Die on error est cochée. |

**Utilisation**

| Règle d’utilisation | Ce composant peut être utilisé en tant que composant standalone dans un Job ou un sous-job. |
Scénario associé

Aucun scénario n’est disponible pour ce composant.
tSSH

Ce composant établit une communication avec un serveur distant et renvoie de manière sécurisée les données sensibles.

Le tSSH renvoie les données d’un ordinateur distant via le protocole Secure Shell défini.

Propriétés du tSSH Standard

Ces propriétés sont utilisées pour configurer le tSSH s’exécutant dans le framework de Jobs Standard. Le composant tSSH Standard appartient à la famille System.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
• View schema : sélectionnez cette option afin de voir le schéma.
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
Cliquez sur Sync columns pour récupérer le schéma du composant précédent dans le Job. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute.</td>
</tr>
<tr>
<td><strong>User</strong></td>
<td>Informations d’authentification de l’utilisateur.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>
| **Public Key/Key Passphrase/Private Key** | Sélectionnez l’option adéquate.  
Si vous choisissez l’option **Public Key**, saisissez la passphrase (phrase secrète), si nécessaire, dans le champ **Key passphrase**. Puis, dans le champ **Private Key**, saisissez la clé privée ou cliquez sur le bouton [...] à côté du champ **Private Key** pour parcourir votre répertoire jusqu’à la clé privée.  
Pour saisir la phrase secrète, cliquez sur le bouton [...] à côté du champ **Passphrase**, puis, dans la boîte de dialogue qui s’ouvre, saisissez la phrase secrète entre guillemets doubles puis cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Password/Password** | Sélectionnez l’option adéquate.  
Si vous choisissez l’option **Password**, saisissez votre mot de passe dans le champ **Password**.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Keyboard Interactive/Password** | Sélectionnez l’option adéquate.  
Si vous choisissez l’option **Keyboard Interactive**, saisissez le mot de passe requis dans le champ **Password**.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Pseudo terminal** | Cochez cette case pour appeler un interpréteur de commandes interactif effectuant les opérations de terminal. |
| **Command separator** | Saisissez le séparateur de commandes requis. Lorsque la case **Pseudo terminal** est cochée, ce champ n’est plus disponible. |
| **Commands** | Saisissez la commande permettant de récupérer l’information de l’ordinateur distant. Lorsque la case **Pseudo terminal** est cochée, ce tableau devient un émulateur de terminal, et chaque ligne du tableau devient une commande. |
| **Use timeout/timeout in seconds** | Définit la durée avant expiration. Un message notifiant l’expiration sera généré si l’actuel temps de réponse dépasse ce délai d’expiration. |
| **Standard Output** | Sélectionnez la destination de la sortie standard à retourner. La sortie peut être retournée vers : |
- la console (**to console**) : la sortie s’affiche dans la console de la vue **Run**.
- les variables globales (**to global variables**) : la sortie est indiquée par la variable globale correspondante.
- la console et les variables globales (**both to console and global variable**) : la sortie est indiquée par les deux moyens.
- une sortie normale (**normal**) : la sortie est une sortie SSH standard.

**Error Output**

Sélectionnez la destination de la sortie d’erreurs à retourner. La sortie peut être retournée vers :
- la console (**to console**) : la sortie s’affiche dans la console de la vue **Run**.
- les variables globales (**to global variables**) : la sortie est indiquée par la variable globale correspondante.
- la console et les variables globales (**both to console and global variable**) : la sortie est indiquée par les deux moyens.
- une sortie normale (**normal**) : la sortie est une sortie SSH standard.

**Advanced settings**

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Global Variables**

**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td><strong>STDOUT</strong></td>
<td>sortie de l’exécution standard de la commande distante. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
<tr>
<td><strong>STDERR</strong></td>
<td>sortie de l’exécution en erreur de la commande distante. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
<tr>
<td><strong>EXIT_CODE</strong></td>
<td>code de sortie de la commande distante. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
**Utilisation**

| Règle d'utilisation | Ce composant peut être utilisé en stand-alone.
|                     | Pour les informations sensibles, telles que le mot de passe et l'identifiant, vous pouvez définir les variables de contexte dans l'onglet *Contexts* et appuyer sur les touches `Ctrl+Espace` pour y accéder.
| Connections         | Liens de sortie (de ce composant à un autre) :
|                     | **Row** : Main.
|                     | Liens d'entrée (d'un autre composant à celui-ci) :
|                     | **Row** : Main, Iterate.
|                     | Pour plus d'informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*.
| Limitation          | L'utilisation de ce composant est adaptée pour les systèmes de type Unix.

**Scénario : Accéder aux informations du système distant via SSH**

Le scénario suivant décrit un Job utilisant un protocole SSH pour afficher le nom d'hôte du serveur distant auquel vous souhaitez vous connecter et la date actuelle de ce système distant.

Un composant **tSSH** est suffisant pour ce Job. Cliquez et déposez-le de la famille *System* de la *Palette* dans l'espace de modélisation graphique.

Double-cliquez sur le **tSSH** et sélectionnez l'onglet *Basic settings* de la vue *Component*.
Procédure

1. Renseignez le nom de l'hôte dans le champ Host auquel accéder à partir du SSH, ainsi que le numéro du port dans le champ Port.
2. Dans le champ User, renseignez le nom d'identification de l'utilisateur sur l'ordinateur distant.
3. Sélectionnez la méthode d'authentification dans la liste déroulante Authentication method. Dans cet exemple, la méthode d'authentification utilisée est Public key (clé publique).
4. Renseignez donc le champ Private key avec votre clé privée.
5. Dans le champ Command, saisissez la commande suivante. Dans cet exemple, saisissez hostname; date entre guillemets doubles (code Java).
6. Cochez la case Use timeout et paramétrez la durée avant expiration à 5 secondes.

Résultats

picasso
Wed Sep 26 14:24:15 CEST 2007
Job uniteElisa ended at 16:26 26/09/2007. [exit code=0]

L'ordinateur distant renvoie le nom de l'hôte, ainsi que la date et l'heure en cours.
tStatCatcher

Ce composant regroupe les métadonnées du Job en cours d’exécution, ainsi que les métadonnées de chaque composant de ce Job. Il collecte et transfère les données de log vers le composant suivant afin qu’elles soient affichées ou stockées.

Le tStatCatcher est basé sur le schéma prédéfini et regroupe les métadonnées du Job en cours d’exécution, ainsi que les métadonnées de chaque composant de ce Job lorsque la case tStatCatcher Statistics est cochée.

Propriétés du tStatCatcher Standard

Ces propriétés sont utilisées pour configurer le tStatCatcher s’exécutant dans le framework de Jobs Standard.

Le composant tStatCatcher Standard appartient à la famille Logs & Errors.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Schema</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Pour ce composant, le schéma est en lecture seule, puisqu’il regroupe les informations de log standard, notamment :</td>
</tr>
<tr>
<td><strong>Moment</strong></td>
<td>Date et heure du traitement</td>
</tr>
<tr>
<td><strong>Pid</strong></td>
<td>Identifiant de processus du Job.</td>
</tr>
<tr>
<td><strong>Father_pid</strong></td>
<td>Identifiant de processus du Job parent, le cas échéant. Sinon, le Pid est dupliqué.</td>
</tr>
<tr>
<td><strong>Root_pid</strong></td>
<td>Identifiant de processus du Job racine, le cas échéant. Sinon, le Pid est dupliqué.</td>
</tr>
<tr>
<td><strong>System_pid</strong></td>
<td>Identifiant du process.</td>
</tr>
<tr>
<td><strong>Project</strong></td>
<td>Nom du projet auquel appartient le Job.</td>
</tr>
<tr>
<td><strong>Job</strong></td>
<td>Nom du Job en cours.</td>
</tr>
<tr>
<td><strong>Job_repository_id</strong></td>
<td>Identifiant du fichier .item du Job stocké dans le référentiel.</td>
</tr>
<tr>
<td><strong>Job_version</strong></td>
<td>Version du Job en cours.</td>
</tr>
<tr>
<td><strong>Context</strong></td>
<td>Nom du contexte en cours.</td>
</tr>
<tr>
<td><strong>Origin</strong></td>
<td>Nom du composant, le cas échéant.</td>
</tr>
<tr>
<td><strong>Message_type</strong></td>
<td>Informations de début ou de fin du Job.</td>
</tr>
</tbody>
</table>
**Message** : Informations de réussite ou d’échec du Job.

**Duration** : Temps d’exécution d’un Job, ou d’un composant si la case tStatCatcher Statistics est cochée.

### Global Variables

**Global Variables**

- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

- Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

- Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

- Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

| Règle d’utilisation | Ce composant est le composant de début d’un Job secondaire qui se déclenche automatiquement à la fin du Job principal. Les dates de début et de fin apparaissent dans le log. |

### Scénario : Afficher les statistiques d’exécution du Job

Ce scénario collecte les statistiques d’exécution du Job et les affiche dans la console Run. Notez que, comme aucune case tStatCatcher Statistics n’a été cochée pour les composants, les statistiques ne s’appliquent qu’à ce Job.

### Relier les composants

**Procédure**

1. Déposez un tFixedFlowInput, un tFileOutputDelimited, un tStatCatcher et un tLogRow dans l’espace de modélisation graphique.

2. Reliez le composant tFixedFlowInput au tFileOutputDelimited à l’aide d’un lien Row > Main.

3. Reliez le composant tStatCatcher au tLogRow à l’aide d’un lien Row > Main.
Configurer les composants

Procédure

1. Double-cliquez sur le tFixedFlowInput pour ouvrir sa vue Basic settings.

2. Cliquez sur le bouton Edit schema pour ouvrir l’éditeur de schéma.

3. Cliquez trois fois sur le bouton [+] pour ajouter trois colonnes, nommées respectivement ID_Owners, Name_Customer et ID_Insurance, de type Integer et String.

4. Cliquez sur OK pour valider la configuration et fermer l’éditeur.

5. Dans la fenêtre qui apparaît, cliquez sur Yes pour accepter la propagation des modifications.
6. Sélectionnez l'option **Use Inline Content (delimited file)**.

7. Dans le champ **Content**, saisissez `1;Andrew;888`.

8. Double-cliquez sur le tFileOutputDelimited pour ouvrir sa vue **Basic settings**.

9. Dans le champ **File Name**, saisissez le chemin d'accès complet du fichier vers lequel vous souhaitez sauvegarder les statistiques.

10. Double-cliquez sur le tLogRow pour ouvrir sa vue **Basic settings**.

11. Sélectionnez l'option **Vertical (each row is a key/value list)** pour un affichage optimisé des résultats.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.

2. Appuyez sur **F6** pour exécuter le Job.
Comme affiché ci-dessus, les statistiques d'exécution du Job ont été correctement générées.
tSugarCRMInput

Ce composant extrait les données d’une base de données SugarCRM à l’aide d’une requête.

Le tSugarCRMInput se connecte à un module de la base de données SugarCRM via le service Web adéquat.

Propriétés du tSugarCRMInput Standard

Ces propriétés sont utilisées pour configurer le tSugarCRMInput s’exécutant dans le framework de Jobs Standard.

Le composant tSugarCRMInput Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>SugarCRM Webservice URL</th>
<th>Saisissez l’URL du service Web permettant de se connecter à la base de données SugarCRM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d’authentification de l’utilisateur au service Web.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Module</td>
<td>Sélectionnez le module adéquat dans la liste.</td>
</tr>
<tr>
<td></td>
<td><strong>Rémarque :</strong></td>
</tr>
<tr>
<td></td>
<td>Pour utiliser des tables personnalisées, sélectionnez Use custom module dans la liste.</td>
</tr>
<tr>
<td></td>
<td>Les champs Custom module package name et Custom module name qui s’affichent sont automatiquement remplis avec les noms correspondants.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans</td>
</tr>
</tbody>
</table>
le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Pour ce composant, le schéma dépend du Module sélectionné.

<table>
<thead>
<tr>
<th>Query condition</th>
<th>Saisissez la requête permettant de sélectionner les données à extraire. Exemple : account_name= 'Talend'.</th>
</tr>
</thead>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
<th>Cochez cette case pour collecter les données de log au niveau du composant.</th>
</tr>
</thead>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire.</th>
</tr>
</thead>
</table>

**Scénario : Extraire les données d’un compte à partir de SugarCRM**

Ce scénario décrit un Job à deux composants visant à extraire les informations d’un compte à partir d’une base de données SugarCRM vers un fichier de sortie Excel.
Configurer le Job

Procédure
1. A partir de la Palette, cliquez-déposez un composant tSugarCRMInput et un composant tFileOutputExcel.
2. Connectez les composants d'entrée et de sortie à l'aide d'un lien de type Row > Main.

Configurer le composant d'entrée

Procédure
1. Double-cliquez sur le composant tSugarCRMInput pour afficher l'onglet Basic settings de la vue Component et paramétrer ses propriétés.

2. Remplissez les informations de connexion dans les champs SugarCRM Web Service URL, Username et Password.
3. Sélectionnez ensuite le module adéquat parmi ceux proposés dans la liste Module. Dans cet exemple, c'est Accounts qui est sélectionné.
   Le champ Schema est automatiquement renseigné en fonction du module sélectionné, mais il est possible de le modifier et de supprimer les colonnes dont vous n'avez pas besoin en sortie.
4. Dans le champ Query Condition, saisissez la requête que vous voulez utiliser pour extraire les données du CRM. Dans cet exemple : "billing_address_city='Sunnyvale'".

Exécuter le Job

Procédure
1. Sélectionnez ensuite le composant tFileOutputExcel.
2. Renseignez le nom du fichier de destination dans le champ **File Name** ainsi que le nom de la feuille de calcul dans le champ **Sheet** puis cochez la case **Include header** pour considérer les entêtes.

3. Enregistrez votre Job puis appuyez sur **F6** pour l’exécuter.

Les données filtrées sont inscrites en sortie dans la feuille de calcul du fichier Excel spécifié.
tSugarCRMOoutput

Ce composant écrit des données dans une base de données SugarCRM.

Le tSugarCRMOoutput écrit dans un module de la base de données SugarCRM via le service Web adéquat.

Propriétés du tSugarCRMOoutput Standard

Ces propriétés sont utilisées pour configurer le tSugarCRMOoutput s’exécutant dans le framework de Jobs Standard.

Le composant tSugarCRMOoutput Standard appartient aux familles Business et Cloud.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>SugarCRM WebService URL</th>
<th>Saisissez l’URL du service Web permettant de se connecter à la base de données SugarCRM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d’authentification de l’utilisateur au service Web.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Module</td>
<td>Sélectionnez le module adéquat dans la liste.</td>
</tr>
<tr>
<td></td>
<td>Remarque :</td>
</tr>
<tr>
<td></td>
<td>Pour utiliser des tables personnalisées, sélectionnez Use custom module dans la liste.</td>
</tr>
<tr>
<td></td>
<td>Les champs Custom module package name et Custom module name qui s’affichent sont</td>
</tr>
<tr>
<td></td>
<td>automatiquement remplis avec les noms correspondants.</td>
</tr>
<tr>
<td>Action</td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données du module SugarCRM :</td>
</tr>
<tr>
<td></td>
<td>Insert : insérer les données.</td>
</tr>
<tr>
<td></td>
<td>Update : mettre les données à jour.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui</td>
</tr>
<tr>
<td></td>
<td>sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le</td>
</tr>
<tr>
<td></td>
<td>mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository,</td>
</tr>
<tr>
<td></td>
<td>trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
</tbody>
</table>
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable **After** et retourne un entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant est utilisé comme composant de sortie. Il nécessite un composant d’entrée. |

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tSVNLogInput

Ce composant récupère les informations d'une révision spécifiée ou d'un ensemble de révisions d'un référentiel SVN.

Propriétés du tSVNLogInput Standard

Ces propriétés sont utilisées pour configurer le tSVNLogInput s'exécutant dans le framework de Jobs Standard.
Le composant tSVNLogInput Standard appartient à la famille Internet.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>URL</th>
<th>Saisissez l'URL du référentiel SVN auquel accéder.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Need authentication?</td>
<td>Cochez cette case si l'authentification est requise pour accéder au référentiel SVN.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez le nom et le mot de passe pour accéder au référentiel SVN si une authentification est nécessaire. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. Ces deux champs sont disponibles uniquement lorsque la case Need authentication? est cochée.</td>
</tr>
<tr>
<td>From revision</td>
<td>Spécifiez un numéro de révision de début pour l'ensemble de révisions.</td>
</tr>
<tr>
<td>Only this revision</td>
<td>Cochez cette case si vous souhaitez récupérer les informations de la seule révision spécifiée dans le champ From revision.</td>
</tr>
<tr>
<td>To revision</td>
<td>Spécifiez un numéro de révision de fin pour l'ensemble de révisions. Saisissez 'HEAD' pour la dernière révision. Le champ disparaît lorsque la case Only this revision est cochée.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit les champs à traiter et à passer au composant suivant. Le schéma de ce composant est en lecture seule. Il décrit les propriétés de la révision. Cliquez sur le bouton [...] à côté du champ Edit schema pour voir le schéma prédéfini contenant les champs suivants : • revision : numéro de révision du message de log. • author : auteur du message de commit. • date : date du message de commit.</td>
</tr>
</tbody>
</table>
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job et de chaque composant. |

### Utilisation

| Règle d'utilisation | Ce composant est généralement utilisé comme composant d’entrée dans un Job et nécessite un lien de sortie. |

| Variables globales | **NB_LINE** : nombre de lignes traitées. Cette variable est une variable *After* et retourne un entier.  

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.  

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

| Connections | Liens de sortie (de ce composant à un autre) :  

**Row** : Main, Iterate  

**Trigger** : On Subjob Ok, On Subjob Error, Run if, On Component Ok, On Component Error  

Liens d’entrée (d’un autre composant à celui-ci) :  

**Row** : Main, Iterate  

**Trigger** : On Subjob Ok, On Subjob Error, Run if, On Component Ok, On Component Error  

Pour plus d’informations concernant les connexions, consultez le *Guide utilisateur du Studio Talend*. |

- **message** : contenu du message.  

- **nb_file_added** : nombre de fichiers ajoutés.  

- **nb_file_modified** : nombre de fichiers modifiés.  

- **nb_file_deleted** : nombre de fichiers supprimés.
**Scénario : Récupérer un message de log depuis un référentiel SVN**

Ce scénario décrit un Job de deux composants récupérant les informations d'une révision spécifiée depuis un référentiel SVN et affiche les informations dans la console.

![Diagramme du Job](image)

**Procédure**

1. Créez un nouveau Job et ajoutez les deux composants suivants en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la Palette : un `tSVNLogInput` et un `tLogRow`.
2. Reliez le `tSVNLogInput` au `tLogRow` à l'aide d'un lien `Row > Main`.
3. Double-cliquez sur le `tSVNLogInput` pour ouvrir sa vue `Basic settings`.
5. Cochez la case `Need authentication?` et renseignez les champs `Username` et `Password` avec vos informations d'authentification.
6. Spécifiez une révision ou un ensemble de révisions. Dans cet exemple, récupérez les informations concernant la révision 17090, saisissez donc `17090` dans le champ `From revision` et cochez la case `Only this revision`.
7. Double-cliquez sur le `tLogRow` pour ouvrir sa vue `Basic settings`. 
8. Définissez la manière dont vous souhaitez afficher les données de sortie dans la console. Dans cet exemple, saisissez "\n" dans le champ Field Separator pour afficher chaque champ d'information dans une nouvelle ligne.

9. Cochez la case Print schema column name in front of each value pour afficher le nom de chaque champ concernant les informations de révision au début de chaque ligne.

10. Appuyez sur les touches Ctrl + S pour sauvegarder le Job puis sur F6 pour l'exécuter.

    [statistics] connecting to socket on port 3665
    [statistics] connected
    revision: 17090
    author: lli
    date: Fri Oct 24 16:03:03 CST 2014
    message: Updated the Properties table for tOracleCDC
    Added a new component tOracleCDCOutput with a scenario
    Updated the chapter file ch-components-databases-traditional.xml

    DOCT-3519, arch used, keywords added, change tracked
    nb_file_added: 16
    nb_file_modified: 2
    nb_file_deleted: 0
    [statistics] disconnected

Comme montré dans la capture d'écran, les informations de la révision spécifiée sont affichées dans la console.
tSybaseBulkExec

Ce composant améliore les performances lors du traitement des données de la base de données Sybase.

Les composants tSybaseOutputBulk et tSybaseBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tSybaseOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Propriétés du tSybaseBulkExec Standard

Ces propriétés sont utilisées pour configurer le tSybaseBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Sélectionnez dans la liste déroulante la version de la base de données Sybase à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Adresse IP du serveur de base de données. Cette option est disponible lorsque vous sélectionnez <strong>Bulk Update</strong> dans la liste <strong>Action on data</strong>, dans la vue <strong>Advanced settings</strong> du tSybaseBulkExec.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données. Cette option est disponible lorsque vous sélectionnez <strong>Bulk Update</strong> dans la liste <strong>Action on data</strong>, dans la vue <strong>Advanced settings</strong> du tSybaseBulkExec.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Sybase Schema</td>
<td>Nom du schéma de la base de données que vous souhaitez utiliser.</td>
</tr>
<tr>
<td>Server</td>
<td>Adresse IP du serveur de base de données pour la connexion de l’utilitaire Bcp.</td>
</tr>
<tr>
<td>Bcp Utility</td>
<td>Nom de l’outil bcp à utiliser pour copier les données sur le serveur Sybase.</td>
</tr>
<tr>
<td>Batch size</td>
<td>Nombre de lignes dans chaque lot de traitement de données.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit déjà exister pour que l’opération d’Insert soit autorisée.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée : <strong>None</strong> : n’effectuer aucune opération sur la table.</td>
</tr>
</tbody>
</table>
### File Name

Nom du fichier à charger.

**Avertissement**:

Ce fichier doit se situer sur la même machine que le serveur de la base de données.

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nomenclature des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

**Repository** : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center ([https://help.talend.com](https://help.talend.com)).

Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].
### Advanced settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use an interface file</strong></td>
<td>Cochez cette case afin de spécifier un fichier interface dans le champ <strong>Interface file</strong>.</td>
</tr>
<tr>
<td><strong>Additional JDBC parameters</strong></td>
<td>Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple : CHARSET=KANJISIS_OS pour supporter les caractères japonais.</td>
</tr>
</tbody>
</table>
| **Action on data**                           | Vous pouvez effectuer les actions suivantes sur la table définie :  

- **Bulk Insert** : ajoute des entrées multiples à la table. Le Job s’arrête lorsque des doublons sont trouvés.  

- **Bulk Update** : effectue des changements simultanés aux multiples entrées.                                                                                                                                                                                                                                                                                                                      |
| **Field Terminator**                         | Caractère, chaîne ou expression régulière séparant les champs.                                                                                                                                                                                                                                                                                                                                                                                                       |
| **Row Terminator**                           | Chaîne (ex : "\n" sous Unix) séparant les lignes.                                                                                                                                                                                                                                                                                                                                                                           |
| **Head row**                                 | Nombre de lignes d’en-tête à ignorer en début de fichier.                                                                                                                                                                                                                                                                                                                                                                                                                |
| **Encoding**                                 | Sélectionnez l’encodage dans la liste ou saisissez-le entre guillemets doubles si le type d’encodage utilisé n’existe pas dans la liste. Ce champ est obligatoire pour la gestion de données de bases de données.                                                                                                                                                                                                                                                                 |
| **Output**                                   | Sélectionnez la sortie à utiliser pour récupérer la sortie standard de la base Sybase :  

- **to console** : vers la console.  

- **to global variable** : vers la variable globale.                                                                                                                                                                                                                                                                                                                                                          |
| **tStatCatcher Statistics**                  | Cochez cette case pour collecter les données de log au niveau du Job, ainsi qu’au niveau du composant.                                                                                                                                                                                                                                                                                                                                                                       |

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE_DATA</strong></td>
<td>nombre de lignes lues. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_BAD</strong></td>
<td>nombre de lignes rejetées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace**.
pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé lorsque les données à charger dans la base n’exigent aucune transformation particulière.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.  


| Limitation          | Le serveur/client de la base de données doit être installé sur la même machine que le Studio Talend ou le Job contenant un tSybaseBulkExec, afin que le composant fonctionne correctement.  

Contrairement au composant Oracle dédié, ce composant Sybase ne permet aucune action directe sur les données.  

Ce composant requiert l’installation des fichiers .jar liés. |

### Scénarios associés

Pour un scénario associé au composant tSybaseBulkExec, consultez :
• Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du tMysqlOutputBulkExec.

• Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du tOracleBulkExec.
**tSybaseClose**

Ce composant ferme une connexion à la base de données Sybase.

**Propriétés du tSybaseClose Standard**

Ces propriétés sont utilisées pour configurer le tSybaseClose s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tSybaseConnection dans la liste.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Scénario associé

Ce composant est étroitement lié aux composants **tSybaseConnection** et **tSybaseRollback**. Il est généralement utilisé avec un composant **tSybaseConnection** car il permet de fermer une connexion pour la transaction en cours.

Pour un scénario associé au composant **tSybaseClose**, consultez **tMysqlConnection** à la page 2618.
tSybaseCommit

Ce composant utilise une connexion unique pour commiter en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

Le tSybaseCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tSybaseCommit Standard

Ces propriétés sont utilisées pour configurer le tSybaseCommit s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tSybaseCommit dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Avertissement :

Si vous utilisez un lien de type Row > Main pour relier le tSybaseCommit à votre Job, vos données seront commitées ligne par ligne.

Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants Sybase, notamment les composants tSybaseConnection et tSybaseRollback.</th>
</tr>
</thead>
</table>
| Dynamic settings    | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  
La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.  
Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend. |

### Scénario associé

Pour plus d’informations relatives au fonctionnement du composant tSybaseCommit, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tSybaseConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Propriétés du tSybaseConnection Standard

Ces propriétés sont utilisées pour configurer le tSybaseConnection s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td>• Built-In : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
<td></td>
</tr>
<tr>
<td>• Repository : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
<td></td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version de la base de données.</td>
</tr>
<tr>
<td>Host</td>
<td>Saisissez l’adresse IP ou le nom d’hôte de la base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute de la base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les données d’authentification de l’utilisateur de la base de données.</td>
</tr>
</tbody>
</table>
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

**Schema**

- Schéma de la base de données.

**Additional JDBC Parameters**

- Spécifiez des paramètres JDBC supplémentaires pour la connexion à la base de données en cours de création.

**Use or register a shared DB Connection**

- Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ **Shared DB Connection Name** qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l'exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

  Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un composant **tRunJob** ayant une de ces options activée fera échouer votre Job.

**Advanced settings**

**Auto Commit**

- Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

  Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.

  Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

**tStatCatcher Statistics**

- Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.
Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec des composants Sybase, notamment les composants tSybaseCommit et tSybaseRollback. |

Scénarios associés

Pour un scénario associé au composant tSybaseConnection, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tSybaseInput

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma.

Le tSybaseInput lit une base de données et en extrait des champs à l’aide de requêtes. Les champs récupérés sont ensuite transmis au composant suivant via une connexion de flux (Main row).

Propriétés du tSybaseInput Standard

Ces propriétés sont utilisées pour configurer le tSybaseInput s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

| Database | Sélectionnez un type de base de données dans la liste et cliquez sur **Apply**. |
| Property type | Peut être **Built-in** ou **Repository**. |
| Built-in : Propriétés utilisées ponctuellement. | |
| Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. | |

Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant.

Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le **Guide utilisateur du Studio Talend**.

Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th><strong>DB Version</strong></th>
<th>Sélectionnez dans la liste déroulante la version de la base de données Sybase à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Sybase Schema</strong></td>
<td>Nom exact du schéma Sybase.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
</tbody>
</table>
**Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Table Name**
Nom de la table à lire.

**Query type et Query**
Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

**Advanced settings**

<table>
<thead>
<tr>
<th>Trim all the String/Char columns</th>
<th>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trim column</td>
<td>Supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Variables globales**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th><strong>NB_LINE</strong> : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>QUERY</strong> : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Sybase.</th>
</tr>
</thead>
</table>
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


| Limitation | Ce composant requiert l’installation des fichiers .jar liés. |

### Scénario associé

Pour un scénario associé, consultez :

- Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520.
tSybaseIQBulkExec

Ce composant charge des données dans une table d’une base de données Sybase, depuis un fichier plat ou une autre table de base de données.

Propriétés du tSybaseIQBulkExec Standard

Ces propriétés sont utilisées pour configurer le tSybaseIQBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseIQBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque** :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th><strong>Database</strong></th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
</table>
| **Property Type** | Sélectionnez la manière de configurer les informations de connexion.  
- **Built-In** : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.  
| **Use an existing connection** | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.  
Lorsque vous réutilisez une connexion à Sybase 16 (SQL Anywhere) existante, créée par un composant tSybaseConnection et chargeant des données depuis un fichier situé côté client, vous devez sélectionner Sybase 16 (SQL Anywhere) dans la liste déroulante **DB Version**, pour afficher la case **Is client file**, puis la cocher.  

**Remarque** : |
Lorsqu'un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

Cette propriété n'est pas disponible lorsque l'option **Sybase IQ 15** est sélectionnée dans la liste déroulante **DB Version**.

<table>
<thead>
<tr>
<th><strong>DB Version</strong></th>
<th>Sélectionnez dans la liste déroulante la version de la base de données Sybase à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Pour Sybase IQ 12 et Sybase IQ 16, le fichier à charger en masse doit être situé sur la même machine que le serveur Sybase IQ.</td>
</tr>
<tr>
<td></td>
<td>• Pour Sybase IQ 15, le fichier à charger en masse peut être situé côté client, toutefois, cela requiert une configuration spécifique du serveur Sybase IQ. Pour plus d'informations, consultez <a href="https://example.com">Sybase IQ client-side load support enhancements</a> (en anglais).</td>
</tr>
<tr>
<td></td>
<td>• Pour Sybase 16 (SQL Anywhere), le fichier à charger en masse peut être situé côté serveur ou client.</td>
</tr>
<tr>
<td></td>
<td>Avant de charger des données dans Sybase 16 (SQL Anywhere), SQL Anywhere 17 doit être installé. Vous pouvez le télécharger <a href="https://example.com">ici</a>. Une fois l'installation terminée, vous devez vérifier la variable système <strong>Path</strong> afin de vous assurer que le chemin d'installation de SQL Anywhere 17 a bien été ajouté. Redémarrez le Studio (voir l'ordinateur, dans certains cas) pour prendre en compte la nouvelle variable <strong>Path</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Saisissez l'adresse IP ou le nom d'hôte de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cette propriété n'est pas disponible lorsque l'option <strong>Sybase IQ 15</strong> est sélectionnée dans la liste déroulante <strong>DB Version</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Port</strong></th>
<th>Saisissez le numéro du port d'écoute de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cette propriété n'est pas disponible lorsque l'option <strong>Sybase IQ 15</strong> est sélectionnée dans la liste déroulante <strong>DB Version</strong>.</td>
</tr>
</tbody>
</table>
### Data Source
Sélectionnez le type de la source de données à utiliser et complétez les informations DNS correspondantes dans le champ affiché à côté. Les types disponibles sont DSN et FILEDSN.

Lorsque l'option FILEDSN est sélectionnée, un bouton [...] est disponible à côté du champ Data Source pour permettre de parcourir le système jusqu’au fichier source de données souhaité.

Cette propriété est disponible uniquement lorsque l’option Sybase IQ 15 est sélectionnée dans la liste déroulante DB Version.

### Database
Saisissez le nom de la base de données.

### Sybase Schema
Schéma de la base de données.

### Username et Password
Saisissez les données d'authentification de l’utilisateur de la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

### Host
Saisissez l'adresse IP ou le nom d'hôte de la base de données.

Cette propriété n’est pas disponible lorsque l’option Sybase IQ 15 est sélectionnée dans la liste déroulante DB Version.

### Port
Saisissez le numéro du port d’écoute de la base de données.

Cette propriété n’est pas disponible lorsque l’option Sybase IQ 15 est sélectionnée dans la liste déroulante DB Version.

### Data Source
Sélectionnez le type de la source de données à utiliser et complétez les informations DNS correspondantes dans le champ affiché à côté. Les types disponibles sont DSN et FILEDSN.

Lorsque l’option FILEDSN est sélectionnée, un bouton [...] est disponible à côté du champ Data Source pour permettre de parcourir le système jusqu’au fichier source de données souhaité.

Cette propriété est disponible uniquement lorsque l’option Sybase IQ 15 est sélectionnée dans la liste déroulante DB Version.

### Table
Saisissez le nom de la table à écrire.

### Action on table
Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :
- **Default** : n’effectuer aucune opération sur la table.
<table>
<thead>
<tr>
<th><strong>Local filename</strong></th>
<th>Spécifiez le chemin du fichier à charger.</th>
</tr>
</thead>
</table>
| **Is client file** | Cochez cette case si vous souhaitez charger les données depuis le côté client.  
Cette propriété est disponible uniquement lorsque l’option Sybase 16 (SQL Anywhere) est sélectionnée dans la liste déroulante **DB Version**. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.  
- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.  
Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| **Use Insert-Location** | Cochez cette case et spécifiez la table de la base de données source de laquelle importer les données dans la table Sybase. |
| **Server** | Saisissez le nom du serveur stockant les données à importer. |
Cette propriété est disponible uniquement lorsque la case **Use Insert-Location** est cochée.

<table>
<thead>
<tr>
<th><strong>Database</strong></th>
<th>Saisissez le nom de la base de données stockant les données à importer. Cette propriété est disponible uniquement lorsque la case <strong>Use Insert-Location</strong> est cochée.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Schema</strong></th>
<th>Saisissez le nom du schéma source. Cette propriété est disponible uniquement lorsque la case <strong>Use Insert-Location</strong> est cochée.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Table</strong></th>
<th>Saisissez le nom de la table source. Cette propriété est disponible uniquement lorsque la case <strong>Use Insert-Location</strong> est cochée.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Use Source Filter</strong></th>
<th>Cochez cette case et spécifiez le filtre de lignes sur la table source. Cette propriété est disponible uniquement lorsque la case <strong>Use Insert-Location</strong> est cochée.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Where Clause</strong></th>
<th>Saisissez la clause WHERE pour filtrer les lignes à importer dans la table Sybase. Cette propriété est disponible uniquement lorsque la case <strong>Use Source Filter</strong> est cochée.</th>
</tr>
</thead>
</table>

| **Columns Mapping** | Renseignez cette table pour spécifier la relation de mapping entre les colonnes de la table source et les colonnes de la table Sybase.  
- **Column** : nom de la colonne de la table Sybase. Par défaut, les champs de la colonne **Column** sont les mêmes que dans le schéma.  
- **Columns in Source Table** : nom de la colonne correspondante dans la table duquel les données seront importées.  
Cette propriété est disponible uniquement lorsque la case **Use Insert-Location** est cochée. |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>Additional JDBC Parameters</strong></th>
<th>Spécifiez des paramètres JDBC supplémentaires pour la connexion à la base de données en cours de création.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Lines terminated by</strong></th>
<th>Saisissez le caractère, la chaîne ou l’expression régulière séparant les lignes. Cette propriété est indisponible lorsque la case <strong>Use Insert-Location</strong> est cochée.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Field terminated by</strong></th>
<th>Saisissez le caractère, le chaîne ou l’expression régulière séparant les champs. Ce composant permet l’utilisation de séparateurs de champs/lignes orientés Sybase, comme \x09.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use enclosed quotes</strong></td>
<td>Cochez cette case pour utiliser des caractères pour entourer les données. Cette propriété est indisponible lorsque la case <strong>Use Insert-Location</strong> ou <strong>Use fixed length</strong> est cochée.</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Use fixed length</strong></td>
<td>Cochez cette case pour déterminer une longueur fixe pour les lignes de données. Cette propriété est indisponible lorsque la case <strong>Use Insert-Location</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Include Header</strong></td>
<td>Cochez cette case si le fichier à charger contient un en-tête. Cette propriété est indisponible lorsque la case <strong>Use Insert-Location</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Blank as null</strong></td>
<td>Cochez cette case pour stocker les Varchar d’entrée de longueur zéro comme NULL, au lieu de blancs dans la base de données durant l’opération de chargement. Cette propriété est indisponible lorsque la case <strong>Use Insert-Location</strong> ou <strong>Use fixed length</strong> est cochée.</td>
</tr>
</tbody>
</table>
| **On file error**      | Sélectionnez l’action que Sybase doit effectuer lorsqu’une erreur survient :  
  - **Rollback (default)** : annuler la transaction en entier.  
  - **Finish** : terminer les insertions déjà complètes et mettre fin à l’opération de chargement.  
  - **Continue** : ignorer l’erreur et continuer l’opération de chargement.  
  Cette propriété est disponible uniquement lorsque l’option **Sybase 16 (SQL Anywhere)** est sélectionnée dans la liste déroulante **DB Version**. |
| **Message log file**   | Chemin vers le fichier MESSAGE LOG sauvegardant les informations de log relatives aux violations des contraintes d’intégrité, aux types de violations, etc. Il doit être spécifié avec la propriété **Row log file**.  
  Cette propriété est disponible uniquement lorsque l’option **Sybase 16 (SQL Anywhere)** est sélectionnée dans la liste déroulante **DB Version**. |
| **Row log file**       | Chemin vers le fichier ROW LOG sauvegardant les informations de log relatives aux données rejetées, aux horodatages de début et de complétion du chargement, etc. Il doit être spécifié avec la propriété **Message log file**.  
  Cette propriété est disponible uniquement lorsque l’option **Sybase 16 (SQL Anywhere)** est sélectionnée dans la liste déroulante **DB Version**. |
<p>| <strong>Location Options</strong>   | Cochez cette case pour afficher les options relatives à l’emplacement. Pour plus d’informations concernant ces... |</p>
<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted Password</td>
<td>Cochez cette case afin de spécifier l’utilisation du cryptage par défaut du mot de passe de la bibliothèque Open Client Library lors de la connexion au serveur source distant. Cette propriété est disponible uniquement lorsque la case <strong>Use Insert-Location</strong> est cochée.</td>
</tr>
<tr>
<td>Use TDS Packet Size</td>
<td>Cochez cette case et, dans le champ <strong>Packet Size</strong> qui s’affiche, spécifiez la taille des paquets TDS en octets. Cette propriété est disponible uniquement lorsque la case <strong>Location Options</strong> est cochée.</td>
</tr>
<tr>
<td>Quoted Identifier</td>
<td>Cochez cette case et, dans la liste déroulante <strong>Status</strong> affichée, spécifiez le paramètre de l’option QUOTED_IDENTIFIER sur le serveur distant. Cette propriété est disponible uniquement lorsque la case <strong>Location Options</strong> est cochée.</td>
</tr>
<tr>
<td>Set Isolation Level</td>
<td>Cochez cette case et, dans la liste <strong>Isolation Level</strong> qui s’affiche, sélectionnez le niveau d’isolation de la connexion au serveur distant. Cette propriété est disponible uniquement lorsque la case <strong>Location Options</strong> est cochée.</td>
</tr>
<tr>
<td>Limit Number</td>
<td>Spécifiez le nombre maximal de lignes à insérer dans la table Sybase IQ. Cette propriété est disponible uniquement lorsque la case <strong>Insert Load Options</strong> est cochée.</td>
</tr>
<tr>
<td>Notify Number</td>
<td>Configurez les notifications pour être averti via un message à chaque fois que le nombre de lignes est inséré dans la table Sybase IQ. Cette propriété est disponible uniquement lorsque la case <strong>Insert Load Options</strong> est cochée.</td>
</tr>
<tr>
<td>Skip Number</td>
<td>Spécifiez le nombre de lignes à ignorer au début de la table source. Cette propriété est disponible uniquement lorsque la case <strong>Insert Load Options</strong> est cochée.</td>
</tr>
</tbody>
</table>
| **Start Row ID** | Spécifiez l'ID de la ligne de la table Sybase IQ à laquelle commencer l'insertion.  

**Remarque** : Sybase IQ 16 ne supporte pas ce paramètre.  

Cette propriété est disponible uniquement lorsque la case **Insert Load Options** est cochée. |
|------------------|-------------------------------------------------------------------------------------------------|

Cette propriété est disponible uniquement lorsque la case **Use Insert-Location** est cochée. |
| **Word Skip Number** | Saisissez combien de fois l'erreur *Words exceeding the maximum permitted word length not supported* doit être ignorée lors du chargement des données dans la table Sybase IQ.  

Cette propriété est disponible uniquement lorsque la case **Insert Load Options** est cochée. |
| **Ignore Constraint** | Cochez cette case et, dans la table qui s'affiche, spécifiez la violation de contrainte d'intégrité à ignorer lors du chargement des données dans la table Sybase IQ, en cliquant sur le bouton [+] pour ajouter autant de lignes que nécessaire, chaque ligne pour une violation de contrainte et configurez la valeur des colonnes suivantes pour chaque violation de contrainte :  

- **Constraint Type** : cliquez dans la cellule et, dans la liste déroulante, sélectionnez le type de violation de contrainte à ignorer.  

- **Number** : saisissez le nombre maximal de violations de contraintes à ignorer avant d'initialiser un Rollback durant un chargement.  

Cette propriété est disponible uniquement lorsque la case **Insert Load Options** est cochée. |
| **Log In File** | Cochez cette case si vous souhaitez enregistrer les informations relatives à la violation de contrainte dans des fichiers.  

Cette propriété est disponible uniquement lorsque la case **Insert Load Options** est cochée. |
| **Message Log** | Spécifiez le chemin d'accès au fichier MESSAGE LOG.  

Cette propriété est disponible uniquement lorsque la case **Log In File** est cochée. |
| **Row Log** | Spécifiez le chemin d'accès au fichier ROW LOG.  

Cette propriété est disponible uniquement lorsque la case **Log In File** est cochée. |
### Log Delimited By
Spécifiez le séparateur des valeurs de données dans le fichier ROW LOG.
Cette propriété est disponible uniquement lorsque la case Log In File est cochée.

### Log Filter
Cochez cette case et, dans la liste Only Log qui s’affiche, sélectionnez le type d’informations relatif aux violations de contraintes à enregistrer.
Cette propriété est disponible uniquement lorsque la case Log In File est cochée.

### tStatCatcher Statistics
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Utilisation

#### Dynamic settings
Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


### Scénario associé

Pour un scénario associé au tSybaseIQBulkExec, consultez :

- Scénario : Charger en masse de données dans une base de données Sybase IQ 12 à la page 4056 du tSybaseIQOutputBulkExec.
- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du tMysqlOutputBulkExec.
• Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du tOracleBulkExec.
tSybaseIQOutputBulkExec

Ce composant améliore les performances lors d'opérations d'Insert dans une base de données Sybase IQ.
Le tSybaseIQOutputBulkExec effectue une action d'Insert sur les données fournies.

Propriétés du tSybaseIQOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tSybaseIQOutputBulkExec s’exécutant dans le framework de Jobs Standard.
Le composant tSybaseIQOutputBulkExec Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Sélectionnez la manière de configurer les informations de connexion.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Built-In</td>
<td>: les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>• Repository</td>
<td>: les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
</tbody>
</table>

Use an existing connection

<table>
<thead>
<tr>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lorsque vous réutilisez une connexion à Sybase 16 (SQL Anywhere) existante, créée par un composant tSybaseConnection et chargeant des données depuis un fichier situé côté client, vous devez sélectionner Sybase 16 (SQL Anywhere) dans la liste déroulante DB Version, pour afficher la case Is client file, puis la cocher.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

Cette propriété n’est pas disponible lorsque l’option Sybase IQ 15 est sélectionnée dans la liste déroulante DB Version.

| DB Version | Sélectionnez dans la liste déroulante la version de la base de données Sybase à utiliser.  
- Pour Sybase IQ 12 et Sybase IQ 16, le fichier à charger en masse doit être situé sur la même machine que le serveur Sybase IQ.  
- Pour Sybase IQ 15, le fichier à charger en masse peut être situé côté client, toutefois, cela requiert une configuration spécifique du serveur Sybase IQ. Pour plus d’informations, consultez Sybase IQ client-side load support enhancements (en anglais).  
- Pour Sybase 16 (SQL Anywhere), le fichier à charger en masse peut être situé côté serveur ou client.  
  Avant de charger des données dans Sybase 16 (SQL Anywhere), SQL Anywhere 17 doit être installé. Vous pouvez le télécharger ici. Une fois l’installation terminée, vous devez vérifier la variable système Path afin de vous assurer que le chemin d’installation de SQL Anywhere 17 a bien été ajouté. Redémarrez le Studio (voire l’ordinateur, dans certains cas) pour prendre en compte la nouvelle variable Path. |

| Host | Saisissez l’adresse IP ou le nom d’hôte de la base de données.  
Cette propriété n’est pas disponible lorsque l’option Sybase IQ 15 est sélectionnée dans la liste déroulante DB Version. |

| Port | Saisissez le numéro du port d’écoute de la base de données.  
Cette propriété n’est pas disponible lorsque l’option Sybase IQ 15 est sélectionnée dans la liste déroulante DB Version. |

| Data Source | Sélectionnez le type de la source de données à utiliser et complétez les informations DNS correspondantes dans le champ affiché à côté. Les types disponibles sont DSN et FILEDSN.  
Lorsque l’option FILEDSN est sélectionnée, un bouton [...] est disponible à côté du champ Data Source pour permettre de parcourir le système jusqu’au fichier source de données souhaité. |
<table>
<thead>
<tr>
<th><strong>SybaseIQOutputBulkExec</strong></th>
<th>Cette propriété est disponible uniquement lorsque l’option <strong>Sybase IQ 15</strong> est sélectionnée dans la liste déroulante <strong>DB Version</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td><strong>Sybase Schema</strong></td>
<td>Schéma de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez les données d’authentification de l’utilisateur de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Saisissez le nom de la table à écrire.</td>
</tr>
<tr>
<td><strong>Action on table</strong></td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default</strong> : n’effectuer aucune opération sur la table.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Drop and create table</strong> : supprimer la table puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Create table</strong> : créer une table qui n’existe pas encore.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Create table if does not exist</strong> : créer la table si elle n’existe pas encore.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Drop table if exists and create</strong> : supprimer la table si elle existe déjà, puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Clear table</strong> : supprimer le contenu de la table.</td>
</tr>
<tr>
<td><strong>File name</strong></td>
<td>Saisissez le nom du fichier à générer et à charger.</td>
</tr>
<tr>
<td><strong>Is client file</strong></td>
<td>Cochez cette case si vous souhaitez charger les données depuis le côté client.</td>
</tr>
<tr>
<td></td>
<td>Cette propriété est disponible uniquement lorsque l’option <strong>Sybase 16 (SQL Anywhere)</strong> est sélectionnée dans la liste déroulante <strong>DB Version</strong>.</td>
</tr>
<tr>
<td><strong>Append the file</strong></td>
<td>Cochez cette case pour ajouter des lignes à la fin des enregistrements.</td>
</tr>
<tr>
<td><strong>Schema et Edit schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réserver line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : Le schéma existe déjà et il est stocké dans le <strong>Repository</strong>. Ainsi, il peut être réutilisé. Voir également le <strong>Guide utilisateur du Studio Talend</strong>.</td>
</tr>
</tbody>
</table>
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

---

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Additional JDBC parameters</strong></th>
<th>Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Fields terminated by</strong></th>
<th>Saisissez le caractère, la chaîne ou l’expression régulière séparant les champs.</th>
</tr>
</thead>
</table>

**Avertissement :**

Ce composant étant une combinaison des tSybaseOutputBulk et tSybaseIQBulkExec, il ne permet pas l’utilisation de séparateurs de champs/lignes orientés Sybase, tels que `\x09`. Pour obtenir l’effet souhaité, (par exemple afficher les champs sous forme de tableau) vous devez utiliser le tSybaseOutputBulk et le tSybaseIQBulkExec ensemble afin de remplacer le tSybaseOutputBulkExec avec le séparateur `\t` utilisé dans le tSybaseOutputBulk et le séparateur `\x09` utilisé dans le tSybaseIQBulkExec.

<table>
<thead>
<tr>
<th><strong>Lines terminated by</strong></th>
<th>Saisissez le caractère, la chaîne ou l’expression régulière séparant les lignes.</th>
</tr>
</thead>
</table>

<p>| <strong>Use enclosed quotes</strong> | Cochez cette case pour utiliser des caractères pour entourer les données. |</p>
<table>
<thead>
<tr>
<th><strong>Include Head</strong></th>
<th>Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Blank as null</strong></td>
<td>Cochez cette case pour stocker les Varchar d’entrée de longueur zéro comme NULL, au lieu de blancs dans la base de données durant l’opération de chargement.</td>
</tr>
</tbody>
</table>
| **On file error** | Sélectionnez l’action que Sybase doit effectuer lorsqu’une erreur survient :  
  • **Rollback** *(default)* : annuler la transaction en entier.  
  • **Finish** : terminer les insertions déjà complètes et mettre fin à l’opération de chargement.  
  • **Continue** : ignorer l’erreur et continuer l’opération de chargement.  
Cette propriété est disponible uniquement lorsque l’option Sybase 16 *(SQL Anywhere)* est sélectionnée dans la liste déroulante **DB Version**. |
| **Message log file** | Chemin vers le fichier MESSAGE LOG sauvegardant les informations de log relatives aux violations des contraintes d’intégrité, aux types de violations, etc. Il doit être spécifié avec la propriété **Row log file**.  
Cette propriété est disponible uniquement lorsque l’option Sybase 16 *(SQL Anywhere)* est sélectionnée dans la liste déroulante **DB Version**. |
| **Row log file** | Chemin vers le fichier ROW LOG sauvegardant les informations de log relatives aux données rejetées, aux horodatages de début et de complétion du chargement, etc. Il doit être spécifié avec la propriété **Message log file**.  
Cette propriété est disponible uniquement lorsque l’option Sybase 16 *(SQL Anywhere)* est sélectionnée dans la liste déroulante **DB Version**. |
| **Encoding** | Sélectionnez l’encodage à partir de la liste ou sélectionnez **Custom** et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Utilisation

| **Dynamic settings** | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez |
Scénario : Charger en masse de données dans une base de données Sybase IQ 12

Ce scénario décrit un Job écrivant des données des données générées par un composant **tRowGenerator** dans un fichier et les charge en masse depuis ce fichier dans une base de données Sybase IQ 12.

### Construire le Job

**Procédure**

1. Créez un nouveau Job et ajoutez un **tRowGenerator**, ainsi qu'un **tSybaseIQOutputBulkExec**, en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le **tRowGenerator** au **tSybaseIQOutputBulkExec** à l'aide d'un lien à **Row > Main**.

###Configurer les composants

**Procédure**

1. Double-cliquez sur le **tRowGenerator** pour ouvrir son éditeur **RowGenerator Editor**.
2. Cliquez sur le bouton [+ ] pour ajouter deux colonnes et configurez leurs propriétés :

3. Dans le champ **Number of Rows for RowGenerator**, saisissez le nombre de lignes à générer. Dans cet exemple, laissez la valeur par défaut, 100.

4. Cliquez sur **OK** pour fermer l’éditeur et cliquez sur **Yes** pour propager les modifications au composant suivant :

5. Double-cliquez sur le **tSybaseIQOutputBulkExec** pour ouvrir sa vue **Basic settings**.

6. Dans les champs **Host, Port, Database, Username et Password**, saisissez les informations nécessaires pour accéder à la base de données Sybase IQ 12.
7. Dans le champ **Table**, saisissez le nom de la table dans laquelle écrire les données. Dans cet exemple, saisissez *staff*.

8. Dans la liste **Action on table**, sélectionnez **Create table if not exists**.

9. Dans le champ **Filename**, saisissez le chemin d’accès complet au fichier contenant les données.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur **Ctrl+S** pour sauvegarder le Job.

2. Appuyez sur la touche **F6** pour exécuter le Job.

   ```
   Starting job usecase_sybase at 14:50 14/11/2012.
   [statistics] connecting to socket on port 3643
   [statistics] connected
   [statistics] disconnected
   Job usecase_sybase ended at 14:50 14/11/2012. [exit code=0]
   ```

3. Dans la console **Sybase Central**, ouvrez la table *staff* afin de vérifier les données :

   ![Sybase Central](image)

   Comme montré ci-dessus, la table est créée et les données sont insérées.

**Scénarios associés**

Pour un scénario associé au composant **tSybaseIQOutputBulkExec**, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du **tMysqlOutputBulk**.

- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du **tMysqlOutputBulkExec**.

- **Scénario : Supprimer et insérer des données dans une base Oracle** à la page 2914 du **tOracleBulkExec**.
tSybaseOutput

Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.
Le tSybaseOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

Propriétés du tSybaseOutput Standard

Ces propriétés sont utilisées pour configurer le tSybaseOutput s’exécutant dans le framework de Jobs Standard.
Le composant tSybaseOutput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez dans la liste déroulante la version de la base de données Sybase à utiliser.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Table</th>
<th>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit déjà exister pour que l’opération d’Insert soit autorisée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Sybase Schema</td>
<td>Nom exact du schéma Sybase.</td>
</tr>
</tbody>
</table>
| Username et Password | Informations d’authentification de l’utilisateur de base de données.  
                     | Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
                     | **Default** : n’effectuer aucune opération de table.  
                     | **Drop and create table** : supprimer la table puis en créer une nouvelle.  
                     | **Create table** : créer une table qui n’existe pas encore.  
                     | **Create table if not exists** : créer la table si nécessaire.  
                     | **Clear table** : supprimer le contenu de la table.  
                     | **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| Turn on identity insert | Cochez cette case pour utiliser votre propre séquence sur les valeurs Identity des données insérées (plutôt que de laisser le serveur SQL choisir les valeurs séquentielles). |
**Action on data**

Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

**Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.

**Update** : Mettre à jour les entrées existantes.

**Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.

**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

**Delete** : Supprimer les entrées correspondantes au flux d’entrée.

**Avertissement :**

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update et Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

---

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

**Advanced settings**

**Commit every**

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

**Additional Columns**

Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.

**Name** : Saisissez le nom de la colonne à modifier ou à insérer.

**SQL expression** : Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.

**Position** : Sélectionnez **Before, Replace** ou **After**, en fonction de l’action à effectuer sur la colonne de référence.

**Reference column** : Saisissez une colonne de référence que le composant tSybaseOutput peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.
## Use field options
Cochez cette case pour personnaliser une requête, surtout lorsqu'il y a plusieurs actions sur les données.

## Enable debug mode
Cochez cette case pour afficher chaque étape du processus de d'écriture dans la base de données.

## Use Batch
Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.

**Remarque :**
Cette case est disponible uniquement si vous avez choisi l'option *Insert, Update, Single Insert Query* ou *Delete* dans le champ *Action on data*.

## Batch Size
Spécifiez le nombre d’enregistrements à traiter dans chaque lot.
Ce champ est disponible uniquement lorsque la case *Use batch mode* est cochée.

## tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE : nombre de lignes traitées. Cette variable est une variable <em>After</em> et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>QUERY : requête traitée. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d'informations concernant les variables, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
<tr>
<td></td>
<td>Une variable <em>Flow</em> fonctionne durant l'exécution d'un composant. Une variable <em>After</em> fonctionne après l'exécution d’un composant.</td>
</tr>
</tbody>
</table>
Utilisation

Règle d'utilisation

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données Sybase. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.

Dynamic settings

Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Limitation

Ce composant requiert l’installation des fichiers .jar liés.

Scénario associé

Pour un scénario associé au composant tSybaseOutput, consultez :

• Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput à la page 2667.
tSybaseOutputBulk

Ce composant prépare le fichier à utiliser comme paramètre dans la requête INSERT servant à alimenter une base de données Sybase.

Les composants tSybaseOutputBulk et tSybaseBulkExec sont généralement utilisés ensemble pour d’une part générer en sortie le fichier qui sera d’autre part utilisé comme paramètre dans l’exécution de la requête SQL énoncée. Cette exécution en deux étapes est unifiée dans le composant tSybaseOutputBulkExec, détaillé dans une section séparée. L’intérêt de proposer deux composants séparés réside dans le fait que cela permet de procéder à des transformations avant le chargement des données dans la base de données.

Écrit un fichier composé de colonnes et basé sur le séparateur défini et sur les standards Sybase.

**Propriétés du tSybaseOutputBulk Standard**

Ces propriétés sont utilisées pour configurer le tSybaseOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>File Name</td>
<td>Nom du fichier à générer.</td>
</tr>
<tr>
<td>Avertissement</td>
<td>Ce fichier est généré sur la machine locale ou dans un dossier partagé du réseau local.</td>
</tr>
<tr>
<td>Append</td>
<td>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</td>
</tr>
</tbody>
</table>
Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Advanced settings

**Row separator**
Chaîne (ex : `\n` sous Unix) séparant les lignes.

**Field separator**
Caractère, chaîne ou expression régulière séparant les champs.

**Avertissement** :
Cette option respecte la syntaxe Java et ne permet pas l'utilisation de séparateurs de lignes/champs orientés Sybase, comme `\x09`.

**Include header**
Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.
**tSybaseOutputBulk**

**Encoding**
Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

### Utilisation

**Règle d’utilisation**
Ce composant est généralement utilisé avec le composant tSybaseBulkExec. Ensemble ils offrent un gain de performance important pour l’alimentation d’une base de données Sybase.

**Limitation**
Ce composant requiert l’installation des fichiers .jar liés.

### Scénario associé

Pour un scénario associé au composant tSybaseOutputBulk, consultez :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du tMysqlOutputBulk.
- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du tMysqlOutputBulkExec.
- Scénario : Supprimer et insérer des données dans une base Oracle à la page 2914 du tOracleBulkExec.
**tSybaseOutputBulkExec**

Ce composant améliore les performances pendant les opérations d’Insert dans une base de données Sybase.

Les composants tSybaseOutputBulk et tSybaseBulkExec sont généralement utilisés ensemble comme deux parties d’un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l’opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tSybaseOutputBulkExec.

**Propriétés du tSybaseOutputBulkExec Standard**

Ces propriétés sont utilisées pour configurer le tSybaseOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</td>
</tr>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant.</td>
</tr>
<tr>
<td></td>
<td>Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
la vue **Basic settings** du composant de connexion créant cette connexion.

2. **Au niveau enfant**, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Sélectionnez dans la liste déroulante la version de la base de données Sybase à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Adresse IP du serveur de base de données. Seuls <code>localhost</code>, <code>127.0.0.1</code> ou l’adresse IP exacte de la machine locale permettent un fonctionnement optimal. Le serveur de la base de données doit être installé sur la même machine que le <strong>Studio Talend</strong> ou que le Job contenant un <code>tSybaseOutputBulkExec</code>.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton <code>[...]</code> à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Bcp Utility</td>
<td>Nom de l’utilitaire bcp à utiliser pour copier les données sur le serveur Sybase.</td>
</tr>
<tr>
<td>Batch row number</td>
<td>Nombre de lignes à traiter dans chaque lot.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois et la table doit déjà exister pour que l’opération d’Insert soit autorisée.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :

  * **None** : n’effectuer aucune opération de table.  
  * **Drop and create the table** : supprimer la table puis en créer une nouvelle.  
  * **Create a table** : créer une table qui n’existe pas encore.  
  * **Create table if doesn’t exist** : créer la table si nécessaire.  
  * **Clear a table** : supprimer le contenu de la table. |
| File Name  | Nom du fichier à générer et à charger.  

⚠️ **Avertissement** :
Ce fichier est généré sur la machine spécifiée par l’URI dans le champ **Server** et doit être sur la même machine que le serveur de la base de données.

<table>
<thead>
<tr>
<th><strong>Append</strong></th>
<th>Cochez cette option pour ajouter des nouvelles lignes à la fin du fichier.</th>
</tr>
</thead>
</table>

**Schema et Edit Schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

**Repository** : Le schéma existe déjà et il est stocké dans le *Repository*. Ainsi, il peut être réutilisé. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type `Integer` ou `Function`, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>Use an interface file</strong></th>
<th>Cochez cette case afin de spécifier un fichier interface dans le champ <strong>Interface file</strong>.</th>
</tr>
</thead>
</table>

**Additional JDBC parameters**

Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple :
CHARSET=KANJISIS_OS pour supporter les caractères japonais.

<table>
<thead>
<tr>
<th>Action on data</th>
<th>Vous pouvez effectuer les actions suivantes sur la table définie :</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bulk Insert</td>
<td>ajoute des entrées multiples à la table. Le Job s'arrête lorsque des doublons sont trouvés.</td>
</tr>
<tr>
<td>Bulk Update</td>
<td>effectue des changements simultanés aux multiples entrées.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field Terminator</th>
<th>Caractère, chaîne ou expression régulière séparant les champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Avertissement</strong></td>
<td>Ce composant étant une combinaison des tSybaseOutputBulk et tSybaseBulkExec, il ne permet pas l'utilisation de séparateurs de champs/lignes orientés Sybase, tels que \x09. Pour obtenir l'effet souhaité, (par exemple afficher les champs sous forme de tableau) vous devez utiliser le tSybaseOutputBulk et le tSybaseBulkExec ensemble afin de remplacer le tSybaseOutputBulkExec, avec le séparateur \t utilisé dans le tSybaseOutputBulk et le séparateur \x09 utilisé dans le tSybaseBulkExec.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DB Row Terminator</th>
<th>Chaîne (ex : &quot;\n&quot; sous Unix) séparant les lignes dans la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>First row NO. of file</td>
<td>Nombre de lignes d'en-tête à ignorer en début de fichier.</td>
</tr>
<tr>
<td>FILE Row Terminator</td>
<td>Caractère, chaîne ou expression régulière séparant les lignes dans le fichier.</td>
</tr>
<tr>
<td>Include Head</td>
<td>Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Sélectionnez l'encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</td>
</tr>
<tr>
<td>Sybase encoding type</td>
<td>Sélectionnez le type d'encodage spécifique à Sybase pour le traitement des données. Ce type d'encodage permet au tSybaseOutputBulkExec de générer correctement la commande Sybase de masse.</td>
</tr>
<tr>
<td>Output</td>
<td>to console : l’information est chargée.</td>
</tr>
<tr>
<td></td>
<td>to global variable : valeurs retournées à partir de fichiers de log.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>
### Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans la base de données.</th>
</tr>
</thead>
</table>

| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.  

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez la page 2641 et la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.  |
|---------------------|-------------------------------------------------------------------------------------------------|

| Limitation | Le serveur/client de la base de données doit être installé sur la même machine que le **Studio Talend** ou que le Job contenant **tSybaseOutputBulkExec**, afin que le composant fonctionne correctement. |

| Limitation | Ce composant requiert l’installation des fichiers .jar liés. |

### Scénarios associés

Pour un scénario associé au composant **tSybaseOutputBulkExec**, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du tMysqlOutputBulk.
- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du tMysqlOutputBulkExec.
- **Scénario : Supprimer et insérer des données dans une base Oracle** à la page 2914 du tOracleBulkExec.
tSybaseRollback

Ce composant annule la transaction commitée dans la base de données Sybase connectée.

Propriétés du tSybaseRollback Standard

Ces propriétés sont utilisées pour configurer le tSybaseRollback s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tSybaseConnection dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Sybase, notamment les composants tSybaseConnection et tSybaseCommit. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez... |
pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez  Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

---

### Scénario associé

Pour un scénario associé au composant **tSybaseRollback**, consultez **Scénario : Annuler l’insertion de données dans des tables mère/fille** à la page 2623.
tSybaseRow

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler).

Le tSybaseRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Propriétés du tSybaseRow Standard

Ces propriétés sont utilisées pour configurer le tSybaseRow s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

| Database | Sélectionnez un type de base de données dans la liste et cliquez sur Apply. |
| Property type | Peut être Built-in ou Repository. |
| Built-in : | Propriétés utilisées ponctuellement. |
| Repository : | Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
la vue **Basic settings** du composant de connexion créant cette connexion.

2. **Au niveau enfant,** utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend.**

<table>
<thead>
<tr>
<th><strong>DB Version</strong></th>
<th>Sélectionnez dans la liste déroulante la version de la base de données Sybase à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Sybase schema</strong></td>
<td>Nom du schéma Sybase.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table name</strong></td>
<td>Nom de la table à lire.</td>
</tr>
<tr>
<td><strong>Turn on identity insert</strong></td>
<td>Cochez cette case pour utiliser votre propre séquence sur les valeurs Identity des données insérées (plutôt que de laisser le serveur SQL choisir les valeurs séquentielles).</td>
</tr>
<tr>
<td><strong>Query type</strong></td>
<td>Peut être Built-in ou Repository.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez la requête appropriée dans le Repository. Le champ <strong>Query</strong> est renseigné automatiquement.</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Saisissez votre requête en faisant particulièrement attention à l’ordre des champs afin qu’ils correspondent à la définition du schéma.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <em>line</em> lors du nommage des champs.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Rejects.

### Advanced settings

| **Additional JDBC parameters** | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case Use an existing connection est décochée dans les Basic settings. |
| **Propagate QUERY’s recordset** | Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste use column.  

**Remarque :**  
Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type Object. Ce composant est généralement suivi du tParseRecordSet. |
| **Use PreparedStatement** | Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau Set PreparedStatement Parameter, définissez les valeurs des paramètres |
représentés par des ‘?’ dans l’instruction SQL définie dans le champ Query de l’onglet Basic settings.

**Parameter Index** : Saisissez la position du paramètre dans l’instruction SQL.

**Parameter Type** : Saisissez le type du paramètre.

**Parameter Value** : Saisissez la valeur du paramètre.

**Remarque** :
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

### Commit every

Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

**Global Variables**

**QUERY** : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.

**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

**Règle d’utilisation**

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez...
dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Limitation**

Ce composant requiert l’installation des fichiers .jar liés.

**Scénario associé**

Pour un scénario associé au composant tSybaseRow, consultez :

- **Scénario : Combiner deux flux pour une sortie sélective** à la page 2706.
- **Procédure du tDBSQLRow**.
- **Scénario : Supprimer et re-générer un index de table MySQL** à la page 2700 du tMysqlRow.
tSybaseSCD

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée. Le tSybaseSCD reflète et traque les modifications d’une table SCD Sybase dédiée.

**Propriétés du tSybaseSCD Standard**

Ces propriétés sont utilisées pour configurer le tSybaseSCD s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseSCD Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. **Au niveau enfant,** utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

   Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend.*

<table>
<thead>
<tr>
<th><strong>DB Version</strong></th>
<th>Sélectionnez dans la liste déroulante la version de la base de données Sybase à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs. Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• <strong>View schema</strong> : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Change to built-in property</strong> : sélectionnez cette option pour passer le schéma en mode <strong>Built-In</strong> et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Update repository connection</strong> : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur <strong>No</strong> et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre <strong>Repository Content</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <em>Guide utilisateur du Studio Talend.</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCD Editor</th>
<th>L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension. Pour plus d’informations, consultez Méthodologie de gestion du SCD à la page 2716.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use memory saving Mode</td>
<td>Cochez cette case pour améliorer les performances du système.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est décochée par défaut, ce qui vous permet de terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>Additional JDBC parameters</th>
<th>Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n’est pas disponible lorsque vous utilisez l’option Use an existing connection dans les Basic settings.</th>
</tr>
</thead>
<tbody>
<tr>
<td>End date time details</td>
<td>Spécifiez la valeur de temps du paramètre de date et heure de fin du SCD au format HH:mm:ss. La valeur par défaut pour ce champ est 12:00:00. Ce champ apparaît uniquement lorsqu’un SCD de Type 2 est utilisé et lorsque Fixed year value est sélectionné pour créer la date de fin du SCD.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
<tr>
<td>Debug mode</td>
<td>Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.</td>
</tr>
</tbody>
</table>

### Variables globales

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est un composant de sortie. Par conséquent, il requiert un composant et une connexion de type Row Main en entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
</tbody>
</table>

| Limitation | Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD. |

### Scénario associé

Pour un scénario associé, consultez **tMysqlSCD** à la page 2712.
tSybaseSCDELT

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table Sybase SCD dédiée.

Le tSybaseSCDELT reflète et traque les modifications d’une table Sybase SCD dédiée.

Propriétés du tSybaseSCDELT Standard

Ces propriétés sont utilisées pour configurer le tSybaseSCDELT s’exécutant dans le framework de Jobs Standard.

Le composant tSybaseSCDELT Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td>Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remarque :</td>
</tr>
<tr>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
<td></td>
</tr>
<tr>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
<td></td>
</tr>
</tbody>
</table>
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>DB Version</th>
<th>Sélectionnez dans la liste déroulante la version de la base de données Sybase à utiliser.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Source table</td>
<td>Nom de la table contenant les données à filtrer.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à écrire. Notez qu’une seule table peut être écrite à la fois pour que l’opération d’insert soit autorisée.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td>None : n’effectuer aucune opération de table.</td>
</tr>
<tr>
<td></td>
<td>Drop and create the table : supprimer la table puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td>Create a table : créer une table qui n’existe pas encore.</td>
</tr>
<tr>
<td></td>
<td>Create table if doesn’t exist : créer la table si nécessaire.</td>
</tr>
<tr>
<td></td>
<td>Clear a table : supprimer le contenu de la table.</td>
</tr>
<tr>
<td></td>
<td>Truncate table : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
</tbody>
</table>
• **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

• **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

| **Built-in** | Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| **Repository** | Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |

**Surrogate Key**
Sélectionnez dans la liste une colonne à utiliser comme clé de substitution.

**Creation**
Sélectionnez la méthode à utiliser pour générer la clé de substitution.
Pour plus d’informations concernant les méthodes de création, consultez **Méthodologie de gestion du SCD** à la page 2716.

**Source Key**
Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l’unicité des données entrantes.

**Use SCD Type 1 fields**
Utilisez le type 1 si vous n’avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.

**Use SCD Type 2 fields**
Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple. Sélectionnez les colonnes du schéma qui servira de référence pour les modifications.

**Start date**
Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ. Vous pouvez sélectionner l’une des colonnes d’entrée du schéma comme date de départ (Start Date) dans la table SCD.

**End Date**
Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne **End Date** a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l’option **Fixed Year value** et saisir une année fictive.

**Log Active Status**
Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut **true** et **false**. Cette colonne permet de repérer facilement le journal actif.
Log versions : Ajoute une colonne à votre schéma SCD pour renseigner le numéro de version du journal.

Advanced settings

Additional JDBC parameters
Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option n’est pas disponible lorsque vous utilisez l’option Use an existing connection dans les Basic settings.

Debug mode
Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.

tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

Global Variables
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation
Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie et une connexion de type Row Main.

Dynamic settings
Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée.
Dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Limitation

Ce composant requiert l’installation des fichiers .jar liés.

---

**Scénario associé**

Pour un scénario associé, consultez **tMysqlSCD** à la page 2712.
**tSybaseSP**

Ce composant appelle une procédure stockée de base de données Sybase.

**Propriétés du tSybaseSP Standard**

Ces propriétés sont utilisées pour configurer le tSybaseSP s'exécutant dans le framework de Jobs Standard.

Le composant tSybaseSP Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

⚠️ **Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <em>Apply</em>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <em>Built-in</em> ou <em>Repository</em>.</td>
</tr>
<tr>
<td><em>Built-in</em></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><em>Repository</em></td>
<td>Sélectionnez le fichier des propriétés du composant. Les champs suivants sont alors pré-remplis.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use an existing connection</th>
<th>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <em>Component List</em> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>⚠️ <strong>Remarque :</strong></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td>1.</td>
<td>Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue <em>Basic settings</em> du composant de connexion créant cette connexion.</td>
</tr>
<tr>
<td>2.</td>
<td>Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux</td>
</tr>
<tr>
<td><strong>DB Version</strong></td>
<td>Sélectionnez dans la liste déroulante la version de la base de données Sybase à utiliser.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d'écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d'authentification de l'utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton […] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. **Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. **Repository** : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<p>| <strong>SP Name</strong> | Saisissez le nom exact de la Procédure Stockée (SP). |
| <strong>Is Function / Return result in</strong> | Cochez cette case, si une seule valeur doit être retournée. |</p>
<table>
<thead>
<tr>
<th>Sélectionnez dans la liste la colonne du schéma sur laquelle est basée la valeur à obtenir.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Timeout Interval</strong></td>
</tr>
</tbody>
</table>
| **Parameters** | Cliquez sur le bouton [+] et sélectionnez dans le champ *Schema Columns* les différentes colonnes nécessaires à la procédure. Notez que le schéma de la SP peut contenir plus de colonnes qu'il n'y a de paramètres utilisés dans la procédure.
Sélectionnez le Type de paramètre :
- **IN** : paramètre d'entrée (Input)
- **OUT** : paramètre de sortie (Output)/valeur retournée
- **IN OUT** : les paramètres d'entrée doivent être retournées sous forme de valeur, même après modifications via la procédure (fonction).
- **RECORDSET** : les paramètres d'entrée doivent être retournées sous forme d'ensemble de valeurs, au lieu d'une valeur unique.

**Remarque :**
Consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620 si vous voulez analyser un ensemble d'enregistrements d'une table de données ou d’une requête SQL.

---

**Advanced settings**

| **Additional JDBC parameters** | Spécifiez des informations supplémentaires de connexion à la base de données créée. Cette option est disponible lorsque la case *Use an existing connection* est décochée dans les **Basic settings**. |
| **Use Multiple SELECT Procedure** | Cochez cette case pour utiliser des procédures contenant plusieurs instructions SELECT. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

---

**Utilisation**

| **Règle d'utilisation** | Ce composant est un composant intermédiaire. Il peut être utilisé comme composant de début. Dans ce cas, seuls les paramètres d'entrée sont autorisés. |
| **Dynamic settings** | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez... |
dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez _Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte_ à la page 2641 et _Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement_ à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le _Guide utilisateur du Studio Talend_.

| Limitation | La syntaxe de la Procédure Stockée doit correspondre à celle de la base de données. Ce composant requiert l'installation des fichiers .jar liés. |

**Scénario associé**

Pour un scénario associé, consultez:

- _Scénario : Récupérer des informations personnelles à l’aide d’une procédure stockée_ à la page 2594.
- _Scénario : Utiliser le tMysqlSP pour trouver le libellé State à l’aide d’une procédure stockée_ à la page 2734.
- _Scénario : Vérifier le format de numéros à l’aide d’une procédure stockée_ à la page 2976.
- _Scénario : Exécuter une procédure stockée à l’aide du tMDMSP_ à la page 2320.

Consultez également _Scénario : Insérer des données dans des tables mère/fille_ à la page 2620 si vous voulez analyser un ensemble d’enregistrements d’une table de données ou d’une requête SQL.
**tSystem**

Ce composant appelle d’autres commandes de traitement, fonctionnant déjà dans un Job plus grand.

**Propriétés du tSystem Standard**

Ces propriétés sont utilisées pour configurer le tSystem s’exécutant dans le framework de Jobs Standard.

Le composant tSystem Standard appartient à la famille System.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Use home directory</th>
<th>Cochez cette case pour changer le nom et le chemin du répertoire dédié.</th>
</tr>
</thead>
</table>
| Use Single Command | Lorsque la commande requise est particulièrement simple, par exemple si un paramètre est utilisé, sans espace, sélectionnez cette option dans le champ **Command**. Dans ce champ, saisissez la commande système simple à exécuter. Notez que la syntaxe n’est pas vérifiée.  

**Avertissement :**  
Sous Windows, les commandes MS-DOS ne permettent pas de passer directement du dossier courant au dossier contenant le fichier à exécuter. Pour exécuter un fichier, vous devez donc utiliser une première commande afin de changer de dossier courant, puis une deuxième commande exécutant le fichier. |
| Use Array Command | Sélectionnez cette option afin d’activer son champ **Command**. Dans ce champ, saisissez la commande système, un paramètre par ligne.  
Par exemple, saisissez la commande suivante avec les espaces consécutifs, pour Linux :  

```
"cp"
"/temp/source.txt"
"/temp/copy to/"
```

<table>
<thead>
<tr>
<th>Standard Output et Error Output</th>
<th>Sélectionnez le type de sortie auquel les données traitées seront transférées.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>to console</strong> : transmet les données vers la console de la vue Run.</td>
</tr>
</tbody>
</table>
### to global variable

Les données sont envoyées dans une variable de sortie liée au composant **tSystem**.

### both to console and to global variable

Les données sont envoyées à la fois vers la console de la vue **Run** et dans une variable de sortie liée au composant **tSystem**.

### normal

Les données sont envoyées au composant suivant.

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

Cliquez sur **Sync columns** pour récupérer le schéma du composant précédent dans le Job.

#### Built-in

Le schéma sera créé et conservé pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

#### Repository

Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé dans divers projets et Job designs. Voir également le **Guide utilisateur du Studio Talend**.

### Environment variables

Cliquez sur le bouton [+ ] pour ajouter les variables d’environnement nécessaires au Job.

- **name**: Renseignez la syntaxe de la nouvelle variable.
- **value**: Entrez une valeur pour ce paramètre en fonction du contexte.

### Advanced settings

#### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.
<table>
<thead>
<tr>
<th>Global Variables</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERROR_MESSAGE</strong> : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
<td></td>
</tr>
<tr>
<td><strong>OUTPUT</strong> : sortie standard d’un processus. Cette variable est une variable After et retourne une chaîne de caractères.</td>
<td></td>
</tr>
<tr>
<td><strong>ERROROUTPUT</strong> : sortie en erreur d’un processus. Cette variable est une variable After et retourne une chaîne de caractères.</td>
<td></td>
</tr>
<tr>
<td><strong>EXIT_VALUE</strong> : code de sortie d’un processus. Cette variable est une variable After et retourne un nombre entier.</td>
<td></td>
</tr>
<tr>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
<td></td>
</tr>
<tr>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.</td>
<td></td>
</tr>
<tr>
<td>Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Utilisation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Règle d’utilisation</strong></td>
<td>Ce composant sert aux entreprises qui utilisent déjà d’autres applications qu’ils souhaitent intégrer à leur flux via Talend.</td>
</tr>
<tr>
<td><strong>Connections</strong></td>
<td>Liens de sortie (de ce composant à un autre) :</td>
</tr>
<tr>
<td></td>
<td><strong>Row</strong> : Main.</td>
</tr>
<tr>
<td></td>
<td><strong>Trigger</strong> : OnSubjobOk, OnSubjobError, Run if.</td>
</tr>
<tr>
<td>Liens d’entrée (d’un autre composant à celui-ci) :</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Row</strong> : Main, Reject, Iterate.</td>
</tr>
<tr>
<td>Pour plus d’informations concernant les liens, consultez la section relative aux types de connexions, dans le Guide utilisateur du Studio Talend.</td>
<td></td>
</tr>
</tbody>
</table>

**Scénario : Echo ’Hello World!’**

Ce scénario est constitué d’un Job à deux composants permettant d’afficher un message dans la console de la vue Run.
Pour reproduire ce scénario, procédez comme suit :

**Procédure**

1. Déposez un composant **tSystem** de la **Palette** dans l’espace de modélisation graphique.
2. Double-cliquez sur le **tSystem** pour ouvrir sa vue **Component**.
3. Sélectionnez l’option **Use Single Command** pour activer le champ **Command** et saisissez "cmd /c echo Hello World!".
4. Dans la liste **Standard Output**, sélectionnez **to both console and global variable**.
5. Appuyez sur **F6** pour exécuter le Job.

**Résultats**

Starting job tSystem_scenario at 17:26 08/10/2009.

[statistics] connecting to socket on port 3961
[statistics] connected
Hello World!
[statistics] disconnected
Job tSystem_scenario ended at 17:26 08/10/2009. [exit code=0]

Le Job exécute une commande `echo` et affiche la sortie dans la console de log de la vue **Run**.
tTeradataClose

Ce composant ferme la connexion à une base de données connectée.

Propriétés du tTeradataClose Standard

Ces propriétés sont utilisées pour configurer le tTeradataClose s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tTeradataConnection dans la liste.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.</td>
</tr>
</tbody>
</table>
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tTeradataCommit

Ce composant utilise une connexion unique pour commiter en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

Le tTeradataCommit valide les données traitées dans un Job à partir d’une base de données connectée.

**Propriétés du tTeradataCommit Standard**

Ces propriétés sont utilisées pour configurer le tTeradataCommit s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant <strong>tTeradataConnection</strong> dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche. <strong>Avertissement :</strong> Si vous utilisez un lien de type <strong>Row &gt; Main</strong> pour relier le <strong>tTeradataCommit</strong> à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case <strong>Close connection</strong> car la connexion sera fermée avant la fin du commit de votre première ligne.</td>
</tr>
</tbody>
</table>

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |
## Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé avec des composants Teradata, notamment les composants <code>tTeradataConnection</code> et <code>tTeradataRollback</code>.</th>
</tr>
</thead>
</table>
| **Dynamic settings** | Cliquez sur le bouton `+` pour ajouter une ligne à la table. Dans le champ `Code`, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.  

La table **Dynamic settings** est disponible uniquement lorsque la case `Use an existing connection` est cochée dans le vue `Basic settings`. Lorsqu’un paramètre dynamique est configuré, la liste `Component List` de la vue `Basic settings` devient inutilisable.  


## Scénario associé

Pour plus d’informations relatives au fonctionnement du composant `tTeradataCommit`, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
**tTeradataConnection**

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Le tTeradataConnection ouvre une connexion à une base de données afin d'effectuer une transaction.

**Propriétés du tTeradataConnection Standard**

Ces propriétés sont utilisées pour configurer le tTeradataConnection s'exécutant dans le framework de Jobs Standard.

Le composant tTeradataConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

ℹ️ **Remarque :**

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Additional JDBC parameters</strong></td>
<td>Ajoutez des informations de connexion supplémentaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple :</td>
</tr>
</tbody>
</table>
## TeradataConnection

| CHARSET=KANJISIS_OS pour supporter les caractères japonais. |

**Remarque :**
Dans ce champ, vous pouvez paramétrer le type d'encodage.

### Use or register a shared DB Connection

Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ *Shared DB Connection Name* qui s'affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l'exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options *Use dynamic job* et *Use an independent process to run subjob* du composant *tRunJob*. Utiliser une connexion partagée avec un composant *tRunJob* ayant une de ces options activée fera échouer votre Job.

### Advanced settings

#### Query band

Cochez cette case pour utiliser la fonctionnalité Teradata Query Banding afin d’ajouter les métadonnées à la requête à traiter, comme le nom de l’utilisateur exécutant la requête. Cela peut vous permettre, par exemple, d’identifier l’origine de la requête.

Une fois la case cochée, la table *Query Band parameters* s’affiche, dans laquelle vous pouvez saisir les informations des métadonnées à ajouter. Cette information prend la forme de paires clé/valeur, par exemple, *DpID* dans la colonne *Key* et *Finance* dans la colonne *Value*.


#### Auto Commit

Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit.
Notre fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

### tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

### Utilisation

#### Règle d’utilisation

Ce composant est généralement utilisé avec des composants Teradata, notamment les composants `tTeradataCommit` et `tTeradataRollback`.

#### Limitation

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

### Scénario associé

Pour un scénario associé au composant `tTeradataConnection`, consultez `tMysqlConnection` à la page 2618.
**tTeradataFastExport**

Ce composant exporte des lots de données d’une table Teradata vers un système client ou vers une base de données plus petite.

Le tTeradataFastExport exporte très rapidement des lots de données volumineux d’une table ou d’une vue Teradata.

**Propriétés du tTeradataFastExport Standard**

Ces propriétés sont utilisées pour configurer le tTeradataFastExport s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataFastExport Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th><strong>Use Commandline</strong></th>
<th>Cochez cette case pour activer le mode invite de commande.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Java API</strong></td>
<td>Cochez cette case pour activer le mode Java API.</td>
</tr>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Execution platform</strong></td>
<td>Sélectionnez le type de système d’exploitation que vous utilisez.</td>
</tr>
<tr>
<td></td>
<td>Cette option est disponible en mode <strong>Use Commandline</strong>.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database name</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark,</td>
</tr>
</tbody>
</table>
évitez le mot réservé `line` lors du nommage des champs.

<table>
<thead>
<tr>
<th><strong>Built-in</strong> : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le <em>Guide utilisateur du Studio Talend</em>.</th>
</tr>
</thead>
</table>

Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Use query**

Cochez cette case afin d’afficher le champ *Query* où saisir l’instruction SQL.

Cette option est disponible en mode *Use Commandline*.

**Query**

Saisissez l’instruction SQL dans le champ *Query*.

**Log database**

Nom de la base de données de log.

Cette option est disponible en mode *Use Commandline*.

**Log table**

Nom de la table de log.

Cette option est disponible en mode *Use Commandline*.

**Script generated folder**

Parcourez votre répertoire et sélectionnez la destination du fichier qui sera créé.

Cette option est disponible en mode *Use Commandline*.

**Exported file**

Nom et chemin d’accès au fichier qui sera créé.

**Field separator**

Caractère, chaîne ou expression régulière séparant les champs.

Cette option est disponible en mode *Use Java API*.

**Row separator**

Chaîne de caractères (par exemple : "\n" sous Unix) pour séparer des champs.

Cette option est disponible en mode *Use Java API*.
**Error file**

Parcourez votre répertoire et sélectionnez la destination du fichier dans lesquels seront enregistrés les messages d’erreur.

Cette option est disponible en mode **Use Java API**.

---

**Advanced settings**

**Output**

Deux options sont disponibles dans la liste :

- **Output error to file** : écrit en sortie l’erreur dans le fichier spécifié dans le champ **Error log** et continue le traitement.

- **Output error to console** : écrit en sortie l’erreur dans la console et arrête le traitement.

Disponible en mode **Use Commandline**.

---

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

---

**Utilisation**

**Règle d’utilisation**

Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Teradata.

**Limitation**

Si vous avez sélectionné le mode **Use Commandline**, vous devez installer le client Teradata sur la machine où se trouve le Job comprenant un **tTeradataFastExport**.

---

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tTeradataFastLoad**

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma.

Le composant tTeradataFastLoad lit une base de données et en extrait des champs à l’aide de requêtes. Les champs récupérés sont ensuite transmis au composant suivant via une connexion de flux (Main row).

**Propriétés du tTeradataFastLoad Standard**

Ces propriétés sont utilisées pour configurer le tTeradataFastLoad s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataFastLoad Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
<tr>
<td>Execute Batch every</td>
<td>Nombre de lignes par lot à charger.</td>
</tr>
<tr>
<td>Die on error</td>
<td>Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row &gt; Rejects.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark,</td>
</tr>
</tbody>
</table>
évitez le mot réservé **line** lors du nommage des champs.

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

### Advanced settings

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Additional JDBC parameters</strong></td>
<td>Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>nombre de lignes traitées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong></td>
<td>nombre de lignes insérées. Cette variable est une variable <strong>After</strong> et retourne un entier.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option. Une variable <strong>Flow</strong> fonctionne durant l’exécution d’un composant. Une variable <strong>After</strong> fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette</td>
</tr>
</tbody>
</table>
liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Teradata.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d'une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tTeradataFastLoadUtility**

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma.

Le tTeradataFastLoadUtility lit une base de données et en extrait des champs à l’aide de requêtes. Les champs récupérés sont ensuite transmis au composant suivant via une connexion de flux (Main row).

**Propriétés du tTeradataFastLoadUtility Standard**

Ces propriétés sont utilisées pour configurer le tTeradataFastLoadUtility s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataFastLoadUtility Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property type</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Execution platform</strong></td>
<td>Sélectionnez le type de système d’exploitation que vous utilisez.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Nom de l’hôte ou l’adresse IP du serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Database name</strong></td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| **Username et Password** | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Table**         | Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. |
**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.


Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.

- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.

- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Script generated folder**

Parcourez votre répertoire et sélectionnez la destination du fichier qui sera créé.

**Load file**

Entrez le nom du fichier à partir duquel vous souhaitez charger les données.

**Field separator**

Caractère, chaîne ou expression régulière séparant les champs.

**Error file**

Parcourez votre répertoire et sélectionnez la destination du fichier dans lesquels seront enregistrés les messages d’erreur.

### Advanced settings

**Define character set**

Spécifiez l’encodage des caractères dont vous avez besoin pour utiliser votre système.

**Check point**

Saisissez la valeur du point de validation.

**Error files**

Saisissez le nom du fichier dans lequel les messages d’erreur sont stockés. Par défaut, le code saisi est *ERRORFILES table_ERR1,ce qui signifie que les deux tables table_ERR1 et table_ERR2 sont utilisées pour enregistrer les messages d’erreur.*

**Return fastload error**

Cochez cette case pour spécifier le code de retour à partir duquel retourner une erreur dans la console.
### Utilisation

| Règle d’utilisation | Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Teradata. |

### Scénario associé

Pour un scénario associé, consultez [Scénario : Insérer des données dans une table d’une base de données Teradata](#) à la page 4169.
tTeradataInput

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma.

Le tTeradataInput lit une base de données et en extrait des champs à l’aide de requêtes. Les champs récupérés sont ensuite transmis au composant suivant via une connexion de flux (Main row).

Propriétés du tTeradataInput Standard

Ces propriétés sont utilisées pour configurer le tTeradataInput s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.

Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td>Built-in</td>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la
### Table name
Parcourez votre système ou saisissez le nom de la table à utiliser.

### Query type et Query
Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

### Advanced settings

<table>
<thead>
<tr>
<th><strong>Additional JDBC parameters</strong></th>
<th>Ajoutez des informations de connexion supplémentaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple : CHARSET=KANJISIS_OS pour supporter les caractères japonais. Cette option est disponible lorsque la case <strong>Use an existing connection</strong> est décochée dans les <strong>Basic settings</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trim all the String/Char columns</strong></td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.</td>
</tr>
<tr>
<td><strong>Trim column</strong></td>
<td>Cochez cette case pour supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.</td>
</tr>
<tr>
<td><strong>Query band</strong></td>
<td>Cochez cette case pour utiliser la fonctionnalité Teradata Query Banding afin d’ajouter les métadonnées à la requête à traiter, comme le nom de l’utilisateur exécutant la requête. Cela peut vous permettre, par exemple, d’identifier l’origine de la requête. Une fois la case cochée, la table <strong>Query Band parameters</strong> s’affiche, dans laquelle vous pouvez saisir les informations des métadonnées à ajouter. Cette information prend la forme de paires clé/valeur, par exemple, DpiID dans la colonne <strong>Key</strong> et Finance dans la colonne <strong>Value</strong>. Cette case permet de générer l’instruction SET QUERY_BAND FOR SESSION avec les paires clé/valeur déclarées dans la table <strong>Query Band parameters</strong>. Pour plus d’informations concernant cette instruction, consultez <a href="http://www.info.teradata.com/HTMLPubs/DB_TTU_14_00/index.html#page/SQL_Reference/B035_1144_111A/End_Locking-Syntax.027.143.html">http://www.info.teradata.com/HTMLPubs/DB_TTU_14_00/index.html#page/SQL_Reference/B035_1144_111A/End_Locking-Syntax.027.143.html</a> (en anglais). Cette case est indisponible lorsque vous avez cochée la case <strong>Using an existing connection</strong>. Dans ce cas, si vous devez utiliser la fonctionnalité <strong>Query band</strong>, configurez-la dans l’onglet <strong>Advanced settings</strong> du composant de connexion Teradata à utiliser.</td>
</tr>
<tr>
<td><strong>tStat Catcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>
## Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE</strong></td>
<td>Nombre de lignes traitées. Cette variable est une variable <em>After</em> et retourne un entier.</td>
</tr>
<tr>
<td><strong>QUERY</strong></td>
<td>Requête traitée. Cette variable est une variable <em>Flow</em> et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable <em>After</em> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <em>Die on error</em> est décochée, si le composant a cette option. Une variable <em>Flow</em> fonctionne durant l'exécution d'un composant. Une variable <em>After</em> fonctionne après l'exécution d'un composant. Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches <strong>Ctrl+Espace</strong> pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

## Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant couvre toutes les requêtes SQL possibles sur les bases de données Teradata.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu'un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>
Limitation

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center ([https://help.talend.com](https://help.talend.com)).

Scénario associé

Pour un scénario associé, consultez :

- **Scénario : Charger des données dans une base de données Teradata** à la page 4152.
- **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520.
tTeradataMultiLoad

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma.

Le tTeradataMultiLoad lit une base de données et en extrait des champs à l’aide de requêtes. Les champs récupérés sont ensuite transmis au composant suivant via une connexion de flux (Main row).

Propriétés du tTeradataMultiLoad Standard

Ces propriétés sont utilisées pour configurer le tTeradataMultiLoad s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataMultiLoad Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Execution platform</td>
<td>Sélectionnez le type de système d’exploitation que vous utilisez.</td>
</tr>
<tr>
<td>Host</td>
<td>Nom de l’hôte ou l’adresse IP du serveur de la base de données.</td>
</tr>
<tr>
<td>Database name</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
</tbody>
</table>
**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.


Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Script generated folder**

Parcourez votre répertoire et sélectionnez la destination du fichier qui sera créé.

**Action to data**

Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

- **Insert** : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
- **Update** : Mettre à jour les entrées existantes.
- **Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Delete** : Supprimer les entrées correspondantes au flux d’entrée.

**Avertissement** :

Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations Update et Delete. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une utilisation de base, ou à partir des options avancées (*Advanced settings*) pour une utilisation optimisée de ces opérations.

**Where condition in case Delete**

Saisissez une condition, qui, si elle est vérifiée, supprimera la ligne.

Ce champ apparaît uniquement lorsque Delete est sélectionné dans la liste déroulante **Action to data**.
| **Load file** | Entrez le nom du fichier à partir duquel vous souhaitez charger les données. |
| **Field separator** | Caractère, chaîne ou expression régulière séparant les champs. |
| **Error file** | Parcourez votre répertoire et sélectionnez la destination du fichier dans lesquels seront enregistrés les messages d’erreur. |

**Advanced settings**

| **Define Log table** | Cochez cette case pour définir la table de log que vous souhaitez utiliser au lieu de celle par défaut, à savoir celle que vous avez définie dans l’onglet **Basic settings**. La syntaxe requise pour définir la table de log est `databasename.logtablename`. |
| **BEGIN LOAD** | Ce champ vous permet de définir votre commande `BEGIN LOAD` pour lancer ou relancer une tâche TPump. Vous pouvez spécifier le nombre de sessions à utiliser, la limite d’erreurs, ainsi que tout autre paramètre nécessaire à l’exécution de la tâche. Pour plus d’informations, consultez la documentation *Teradata MultiLoad Reference*. |
| **Return mload error** | Cochez cette case pour spécifier le code de retour à partir duquel retourner une erreur dans la console. |
| **Define character set** | Spécifiez l’encodage des caractères dont vous avez besoin pour utiliser votre système. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| **Global Variables** | **EXIT_VALUE** : code de sortie du processus. Cette variable est une variable *After* et retourne un nombre entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |

| **Global Variables** | **EXIT_VALUE** : code de sortie du processus. Cette variable est une variable *After* et retourne un nombre entier.  
**ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

**Règle d’utilisation**

Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Teradata.

**Scénario associé**

Pour un scénario associé, consultez Scénario : Insérer des données dans une table d’une base de données Teradata à la page 4169.
tTeradataOutput

Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.
Le tTeradataOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

Propriétés du tTeradataOutput Standard

Ces propriétés sont utilisées pour configurer le tTeradataOutput s’exécutant dans le framework de Jobs Standard.
Le composant tTeradataOutput Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :
Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.
Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-In ou Repository.</td>
</tr>
<tr>
<td></td>
<td>Built-In : propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td>Repository : sélectionnez le fichier dans lequel sont stockées les propriétés du composant.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td></td>
<td>Remarque :</td>
</tr>
<tr>
<td></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.</td>
</tr>
<tr>
<td></td>
<td>2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.</td>
</tr>
</tbody>
</table>
Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| **Username et Password** | Informations d'authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Table** | Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois. |
| **Action on table** | **Remarque :**  
La liste Action on table est indisponible si vous cochez la case Enable parallel execution dans la vue Advanced settings.  
Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
- **None** : n’effectuer aucune opération de table.  
- **Drop and create the table** : supprimer la table puis en créer une nouvelle.  
- **Create a table** : créer une table qui n’existe pas encore.  
- **Create table if doesn’t exist** : créer la table si nécessaire.  
- **Clear a table** : supprimer le contenu de la table.  
- **Truncate table** : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| Create | Elle n’est pas visible par défaut, sauf si vous choisissez de créer une table à partir de la liste déroulante Action on table. La table à créer peut être :  
- **SET TABLE** : table ne permettant pas de dupliquer les lignes.  
- **MULTI SET TABLE** : table permettant de dupliquer les lignes. |
| **Action on data** | Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée : |
**Insert** : ajoute de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.

**Update** : met à jour les entrées existantes.

**Insert or update** : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.

**Update or insert** : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.

**Delete** : supprime les entrées correspondantes au flux d’entrée.

⚠️ **Avertissement** :
Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton [...] à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

---

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs.

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.

**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center ([https://help.talend.com](https://help.talend.com)).
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre **[Repository Content]**.

**Die on error**

Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

**Additional JDBC parameters**

Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple : **CHARSET=KANJISIS_OS** pour supporter les caractères japonais. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.

**Remarque :**

Vous pouvez appuyer sur **Ctrl+Espace** afin d'accéder à une liste de variables globales prédéfinies.

**Commit every**

Nombre de lignes à inclure dans le lot avant de commencer l'écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d'exécution.

Cette option n'est pas disponible si vous avez coché la case **Use an existing connection** dans la vue **Basic settings**.

**Remarque :**

Si vous avez sélectionné **Drop and create table, Create table, Create table if does not exist** ou **Drop table if exists and create** dans la liste **Action on table**, dans la vue **Basic settings**, saisissez 0 dans le champ afin d'assurer la validité des instructions SQL. Pour plus d'informations concernant la validité des instructions SQL dans une base de données.
<table>
<thead>
<tr>
<th><strong>TeradataOutput</strong></th>
<th><strong>Teradata, consultez info.teradata.com (en anglais).</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Additional Columns</strong></td>
<td>Cette option n'est pas disponible si vous venez de créer la table de données (que vous l'ayez préalablement supprimée ou non). Cette option vous permet d'effectuer des actions sur les colonnes, à l'exclusion des actions d'insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Saisissez le nom de la colonne à modifier ou à insérer.</td>
</tr>
<tr>
<td><strong>SQL expression</strong></td>
<td>Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes.</td>
</tr>
<tr>
<td><strong>Position</strong></td>
<td>Sélectionnez <strong>Before</strong>, <strong>Replace</strong> ou <strong>After</strong>, en fonction de l'action à effectuer sur la colonne de référence.</td>
</tr>
<tr>
<td><strong>Reference column</strong></td>
<td>Saisissez une colonne de référence que le composant <strong>tTeradataOutput</strong> peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier.</td>
</tr>
<tr>
<td><strong>Query band</strong></td>
<td>Cochez cette case pour utiliser la fonctionnalité Teradata Query Banding afin d’ajouter les métadonnées à la requête à traiter, comme le nom de l’utilisateur exécutant la requête. Cela peut vous permettre, par exemple, d’identifier l’origine de la requête. Une fois la case cochée, la table <strong>Query Band parameters</strong> s’affiche, dans laquelle vous pouvez saisir les informations des métadonnées à ajouter. Cette information prend la forme de paires clé/valeur, par exemple, <strong>DpID</strong> dans la colonne <strong>Key</strong> et <strong>Finance</strong> dans la colonne <strong>Value</strong>. Cette case permet de générer l’instruction SET QUERY_BAND FOR SESSION avec les paires clé/valeur déclarées dans la table <strong>Query Band parameters</strong>. Pour plus d’informations concernant cette instruction, consultez <a href="http://www.info.teradata.com/HTMLPubs/DB_TTU_14_00/index.html#page/SQL_Reference/B035_1144_111A/End_Logging-Syntax.027.143.html">http://www.info.teradata.com/HTMLPubs/DB_TTU_14_00/index.html#page/SQL_Reference/B035_1144_111A/End_Logging-Syntax.027.143.html</a> (en anglais). Cette case est indisponible lorsque vous avez coché la case <strong>Using an existing connection</strong>. Dans ce cas, si vous devez utiliser la fonctionnalité <strong>Query band</strong>, configurez-la dans l’onglet <strong>Advanced settings</strong> du composant de connexion Teradata à utiliser.</td>
</tr>
<tr>
<td><strong>Use field options</strong></td>
<td>Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données.</td>
</tr>
<tr>
<td><strong>Enable debug mode</strong></td>
<td>Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.</td>
</tr>
</tbody>
</table>
Use Batch | Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.

Batch Size | Spécifiez le nombre d’enregistrements à traiter dans chaque lot.
Ce champ est disponible uniquement lorsque la case Use batch mode est cochée.

tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

Global Variables | NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.
                 | NB_LINE_UPDATED : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.
                 | NB_LINE_INSERTED : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.
                 | NB_LINE_DELETED : nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.
                 | NB_LINE_REJECTED : nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.
                 | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
                 Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
                 Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
                 Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation | Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données Teradata. Il permet aussi de créer un flux de rejet avec un lien Row > Reject filtrant les données en erreur.

Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de
contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

**Limitation**

Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com).

**Scénario associé**

Pour un scénario associé, consultez :

- **Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput** à la page 2667.
**tTeradataRollback**

Ce composant annule le commit d’une transaction dans la base de données Teradata connectée.

**Propriétés du tTeradataRollback Standard**

Ces propriétés sont utilisées pour configurer le tTeradataRollback s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant <strong>tTeradataConnection</strong> dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants Teradata, notamment les composants **tTeradataConnection** et **tTeradataCommit**. |
| Dynamic settings | Cliquez sur le bouton **[+]** pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez... |
La table **Dynamic settings** est disponible uniquement lorsque la case **Use an existing connection** est cochée dans la vue **Basic settings**. Lorsqu’un paramètre dynamique est configuré, la liste **Component List** de la vue **Basic settings** devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez **Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte** à la page 2641 et **Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement** à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le **Guide utilisateur du Studio Talend**.

### Scénario associé

Pour un scénario associé au composant **tTeradataRollback**, consultez **Scénario : Annuler l’insertion de données dans des tables mère/fille** à la page 2623.
tTeradataRow

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler), selon la nature de la requête et de la base de données.

Le tTeradataRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Propriétés du tTeradataRow Standard

Ces propriétés sont utilisées pour configurer le tTeradataRow s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique.
Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>
| Remarque : | Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :
| 1. | Au niveau parent, enregistrer la connexion à la base de données à partager, dans |
la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
<td></td>
</tr>
<tr>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
<td></td>
</tr>
<tr>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
<td></td>
</tr>
<tr>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Nom de la table à traiter.</td>
</tr>
</tbody>
</table>
### Query type

<table>
<thead>
<tr>
<th></th>
<th>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in</td>
<td>Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez la requête appropriée dans le Repository. Le champ Query est renseigné automatiquement.</td>
</tr>
</tbody>
</table>

### Query

Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs correspond à celui défini dans le schéma.

### Die on error

Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

#### Additional JDBC parameters

Ajoutez des informations de connexion supplémentaires nécessaires à la connexion à la base de données, afin de supporter des caractères spécifiques. Exemple : **CHARSET=KANJISIS_OS** pour supporter les caractères japonais. Cette option est disponible lorsque la case **Use an existing connection** est décochée dans les **Basic settings**.

#### Propagate QUERY’s recordset

Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.

**Remarque :**
Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type **Object**. Ce composant est généralement suivi du **tParseRecordSet**.

#### Use PreparedStatement

Cochez cette case pour utiliser une instance PreparedStatement afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des "?" dans l’instruction SQL définie dans le champ **Query** de l’onglet **Basic settings**.

- **Parameter Index** : Saisissez la position du paramètre dans l’instruction SQL.
- **Parameter Type** : Saisissez le type du paramètre.
- **Parameter Value** : Saisissez la valeur du paramètre.

**Remarque :**
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.

<table>
<thead>
<tr>
<th>Commit every</th>
<th>Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.</th>
</tr>
</thead>
</table>
| Query band | Cochez cette case pour utiliser la fonctionnalité Teradata Query Banding afin d’ajouter les métadonnées à la requête à traiter, comme le nom de l’utilisateur exécutant la requête. Cela peut vous permettre, par exemple, d’identifier l’origine de la requête.  
Une fois la case cochée, la table **Query Band parameters** s’affiche, dans laquelle vous pouvez saisir les informations des métadonnées à ajouter. Cette information prend la forme de paires clé/valeur, par exemple, **DpID** dans la colonne **Key** et **Finance** dans la colonne **Value**.  
Cette case est indisponible lorsque vous avez cochée la case **Using an existing connection**. Dans ce cas, si vous devez utiliser la fonctionnalité **Query band**, configurez-la dans l’onglet **Advanced settings** du composant de connexion Teradata à utiliser. |
| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.  
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.  
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
|---|---|
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

| Limitation | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com). |

Scénario associé

Pour un scénario associé, consultez :

- Scénario : Charger des données dans une base de données Teradata à la page 4152.
- Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
• Procédure du composant **tDBSQLRow**.

• Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du composant **tMysqlRow**.
tTeradataSCD

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table SCD dédiée. Le tTeradataSCD reflète et traque les modifications d’une table Teradata SCD dédiée.

**Propriétés du tTeradataSCD Standard**

Ces propriétés sont utilisées pour configurer le tTeradataSCD s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataSCD Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Peut être <strong>Built-In</strong> ou <strong>Repository</strong>.</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td><strong>Remarque</strong></td>
<td>Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue <strong>Basic settings</strong> du composant de connexion créant cette connexion.</td>
</tr>
</tbody>
</table>
| **2.** Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.  
Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend. |

| **Host** | Saisissez l’adresse IP du serveur de base de données. |
| **Database** | Saisissez le nom de la base de données à utiliser. |
| **Username et Password** | Saisissez les informations d’authentification de l’utilisateur de la base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Table** | Saisissez le nom de la table à écrire. Notez que seule une table peut être écrite à la fois. |
| **Action on table** | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
• **None** : n’effectuer aucune opération de table.  
• **Create table** : créer une table qui n’existe pas encore.  
• **Create table if not exists** : créer la table si nécessaire. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réserve line lors du nommage des champs.  
**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  
**Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.  
Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
• **View schema** : sélectionnez cette option afin de voir le schéma.  
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez |
propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**SCD Editor**

L’éditeur SCD Editor permet de construire et de configurer les données du flux de sortie vers la table Slowly Changing Dimension.

Pour plus d’informations, consultez Méthodologie de gestion du SCD à la page 2716.

**Use memory saving mode**

Cochez cette case pour maximiser les performances du système.

**Source keys include Null**

Cochez cette case pour autoriser les valeurs nulles dans les colonnes de clé source.

⚠️ Avertissement :

Une attention particulière doit être portée à l’unicité des valeurs des clés source lorsque cette option est sélectionnée.

**Die on error**

Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur. Lorsque les erreurs sont ignorées, vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien Row > Reject.

**Advanced settings**

**Additional JDBC Parameters**

Spécifiez les propriétés supplémentaires de connexion pour la connexion à la base de données en cours de création.

Ce champ n’est pas visible lorsque la case Use an existing connection est cochée.

**Debug mode**

Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données.

**tStatCatcher Statistics**

Cochez cette case pour collecter les données de log au niveau du composant.

**Variables globales**

**Global Variables**

- **NB_LINE_UPDATED** : nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.
- **NB_LINE_INSERTED** : nombre de lignes insérées. Cette variable est une variable After et retourne un entier.
- **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.
Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est généralement utilisé comme composant de sortie. Il nécessite un composant d’entrée et un lien <strong>Row &gt; Main</strong> en entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD.</td>
</tr>
</tbody>
</table>

### Scénario associé

Pour un scénario similaire utilisant une base de données MySQL, consultez *Scénario : Identifier des modifications de données en utilisant les dimensions à évolution lente (SCD) de type 0 à 3* à la page 2718.
tTeradataSCDELT

Ce composant répond à des besoins en transformation Slowly Changing Dimension, en lisant régulièrement une source de données et en répertoriant les modifications dans une table Teradata SCD dédiée.

Le tTeradataSCDELT reflète et traque les modifications d’une table Teradata SCD dédiée.

Propriétés du tTeradataSCDELT Standard

Ces propriétés sont utilisées pour configurer le tTeradataSCDELT s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataSCDELT Standard appartient aux familles Business Intelligence et Databases. Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
</table>
| Property Type | Peut être Built-In ou Repository.  
• Built-In : Aucune propriété n’est stockée de manière centrale.  
• Repository : Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà défini.  
Remarque :  
Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :  
1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion. |
2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d'informations concernant le partage d'une connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>Host</th>
<th>Saisissez l’adresse IP ou le nom d’hôte du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Source table</td>
<td>Saisissez le nom de la table d’entrée SCD Teradata.</td>
</tr>
<tr>
<td>Table</td>
<td>Saisissez le nom de la table à écrire. Notez que seule une table peut être écrite à la fois.</td>
</tr>
</tbody>
</table>
| Action on table | Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :  
  - None : n’effectuer aucune opération de table.  
  - Create table : créer une table qui n’existe pas encore.  
  - Drop and create table : supprimer la table puis en créer une nouvelle.  
  - Create table if not exists : créer la table si nécessaire.  
  - Drop table if exists and create : supprimer la table si nécessaire puis la recréer.  
  - Clear table : supprimer le contenu de la table, avec possibilité de Rollback.  
  - Truncate table : supprimer rapidement le contenu de la table, mais sans possibilité de Rollback. |
| Create | Sélectionnez le type de table à créer :  
  - SET TABLE : créer une table SET qui n’admet pas l’insertion de lignes dupliquées.  
  - MULTISET TABLE : créer une table MULTISET qui admet l’insertion de lignes dupliquées.  

Cette liste est seulement disponible si l’option Create table, Drop and create table, Create table if not exists, ou Drop table if exists and create est sélectionnée dans la liste déroulante Action on table. |
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark,
évitez le mot réservé *line* lors du nommage des champs.

- **Built-In**: Le schéma est créé et conservé pour ce composant seulement. Consultez également le *Guide utilisateur du Studio Talend*.

Cliquez sur *Edit schema* pour modifier le schéma. Si le schéma est en mode *Repository*, trois options sont disponibles :

- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Surrogate key
Sélectionnez dans la liste une colonne à utiliser comme clé de substitution.

### Creation
Sélectionnez la méthode à utiliser pour générer la clé de substitution.
Pour plus d’informations concernant les méthodes de création, consultez *Méthodologie de gestion du SCD* à la page 2716.

### Source Keys
Sélectionnez une colonne ou plus à utiliser en tant que clé(s) pour assurer l’unicité des données entrantes.

### Use SCD type 1 fields
Utilisez le type 1 si vous n’avez pas besoin de traquer les modifications, pour des corrections typographiques par exemple.

### SCD type 1 fields
Cliquez sur le bouton [*+*] pour ajouter autant de lignes que nécessaire et pour chaque ligne, sélectionnez une colonne de schéma d’entrée qui sera vérifiée pour les changements de type 1.
Cette table est seulement disponible si la case *Use SCD type 1 fields* est cochée.

### Use SCD type 2 fields
Utilisez le type 2 si vous avez besoin de traquer les modifications, pour garder une trace des mises à jour effectuées par exemple.
SCD type 2 fields

Cliquez sur le bouton [+ ] pour ajouter autant de lignes que nécessaire et pour chaque ligne, sélectionnez une colonne de schéma d’entrée qui sera vérifiée pour les changements de type 2.
Cette table est seulement disponible si la case Use SCD type 2 fields est cochée.

Start date

Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de départ.
Cette liste est seulement disponible si la case Use SCD type 2 fields est cochée.

End date

Ajoute une colonne à votre schéma SCD pour déterminer la valeur de la date de fin pour le journal. Lorsque le journal est en mode actif, la colonne End Date a une valeur nulle ; pour éviter cela, vous pouvez sélectionner l’option Fixed Year value et saisir une année fictive.
Cette liste est seulement disponible si la case Use SCD type 2 fields est cochée.

Log active status

Ajoute une colonne à votre schéma SCD pour renseigner les valeurs de statut true et false. Cette colonne permet de repérer facilement le journal actif.
Cette option est seulement disponible si la case Use SCD type 2 fields est cochée.

Log versions

 Sélectionnez cette case et dans la liste Version field qui s‘affiche, ajoutez une colonne de schéma d‘entrée à votre schéma SCD pour renseigner le numéro de version du journal.
Cette option est seulement disponible si la case Use SCD type 2 fields est cochée.

Advanced settings

Additional JDBC Parameters

Spécifiez des informations supplémentaires de connexion à la base de données créée.
Cette option n’est pas disponible si la case Use an existing connection est cochée.

Source fields value include Null

Cochez cette case pour permettre aux colonnes sources de contenir des valeurs nulles. Les colonnes sources mentionnées ici désignent les champs définis dans les tables SCD type 1 fields et SCD type 2 fields.

 tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement de Job au niveau du Job et au niveau des composants.

Global Variables

Global Variables

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est
une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant est utilisé comme composant de sortie et nécessite toujours un lien d’entrée.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dynamic settings</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ <strong>Code</strong>, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend. La table <strong>Dynamic settings</strong> est disponible uniquement lorsque la case <strong>Use an existing connection</strong> est cochée dans la vue <strong>Basic settings</strong>. Lorsqu’un paramètre dynamique est configuré, la liste <strong>Component List</strong> de la vue <strong>Basic settings</strong> devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez <a href="#">Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte</a> à la page 2641 et <a href="#">Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement</a> à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le <em>Guide utilisateur du Studio Talend</em>.</td>
</tr>
<tr>
<td><strong>Limitation</strong></td>
<td>Ce composant requiert l’installation des fichiers .jar liés.</td>
</tr>
</tbody>
</table>
Scénario associé

Pour des scénarios associés, consultez tMysqlSCD à la page 2712 et Scénario : Identifier des modifications de données en utilisant les dimensions à évolution lente (SCD) de type 0 à 3 à la page 2718.
**tTeradataTPTExec**

Ce composant offre de hautes performances lors de l’insertion des données d’un fichier existant dans une table d’une base de données Teradata.

Le tTeradataTPTExec est une combinaison du tTeradataFastLoad, du tTeradataMultiLoad, du tTeradataTPump et du tTeradataFastExport. Il charge les données d’un fichier existant dans une base de données Teradata.

**Propriétés du tTeradataTPTExec Standard**

Ces propriétés sont utilisées pour configurer le tTeradataTPTExec s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataTPTExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être <strong>Built-In</strong> ou <strong>Repository</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-In</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier de propriétés du composant. Les champs suivants concernant la connexion à la base de données sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Execution platform</strong></td>
<td>Sélectionnez le type de système d’exploitation utilisé pour exécuter le Job, <strong>Windows</strong> ou <strong>Unix</strong>.</td>
</tr>
<tr>
<td><strong>TDPID</strong></td>
<td>Spécifiez le Director program identifier de Teradata. Cela peut-être le nom ou l’adresse IP de la base de données Teradata à laquelle vous accédez.</td>
</tr>
<tr>
<td><strong>Database name</strong></td>
<td>Saisissez dans ce champ le nom de la base de données Teradata.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Spécifiez l’identifiant et mot de passe de l’utilisateur de la base de données Teradata. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Schema et Edit Schema</strong></td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <strong>line</strong> lors du nommage des champs.</td>
</tr>
</tbody>
</table>

**Remarque :**
Il est recommandé de ne pas utiliser le mot-clé Teradata Database comme nom de colonne de base de données. Si vous y êtes obligé, le nom de la colonne de base de données doit alors être entouré par \". Par exemple, lorsque le mot-clé *id* est utilisé comme nom de colonne d’une base de données, le champ **Db Column** doit être renseigné par \"id\".

| **Built-In** | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| **Repository** | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |

Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Consumer Operator**

Sélectionnez un opérateur dans la liste déroulante.

- **Load** : charge un grand volume de données dans une table vide dans la base de données Teradata à l’aide de l’utilitaire Teradata FastLoad utility.
- **Inserter** : insère les données dans une table définie dans la base de données Teradata.
- **Update** : réalise des insertions, des mises à jour ou des suppressions dans une table nouvellement créée ou déjà existante de la base de données Teradata à l’aide de l’utilitaire Teradata MultiLoad utility.
- **Stream** : réalise des insertions, des mises à jour ou des suppressions dans une table nouvellement créée ou déjà existante de la base de données Teradata à l’aide de l’utilitaire Teradata TPump.

### Action on data

Sélectionnez dans la liste déroulante une action à effectuer sur les données.

- **Insert**: ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.
- **Update**: mettre à jour des entrées déjà existantes dans la table.
- **InsertOrUpdate**: ajouter de nouvelles entrées. Une mise à jour sera effectuée si l'entrée avec la référence donnée existe déjà.
- **Delete**: Supprimer des entrées correspondant au flux d'entrée.

Ce champ apparaît uniquement lorsque l'opérateur **Update** ou **Stream** est sélectionné dans la liste déroulante **Consumer Operator**.

### Producer Operator

Sélectionnez un opérateur dans la liste déroulante. Seul l'opérateur **DataConnector** est actuellement supporté.


### Table

Spécifiez le nom de la table à écrire dans la base de données Teradata. Notez que vous ne pouvez écrire qu'une table à la fois.

### Script generated folder

Spécifiez le répertoire sous lequel le fichier script Teradata Parallel Transporter sera créé durant l'exécution du Job puis supprimé avant la fin du Job.

### Load file

Spécifiez le fichier contenant les données à charger dans la base de données Teradata.

### Error file

Spécifiez le fichier dans lequel les messages de log seront enregistrés.

### Advanced settings

#### Field separator

Caractère, chaîne ou expression régulière séparant les champs.

#### Define Log table

Cochez cette case pour spécifier le nom de table de log.

#### Set Script Parameters

Cochez cette case pour spécifier la valeur des paramètres **Load Operator, Data Connector, Job Name, and Layout Name (schema)** qui seront utilisés lors de la génération d’un script durant l’exécution du Job. Si vous ne les spécifiez pas manuellement, le système utilisera leurs valeurs par défaut.

#### Load Operator

Spécifiez l’opérateur de chargement.
Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée.

| Data Connector | Spécifiez le connecteur de données.  
|               | Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée. |

| Job Name | Spécifiez le nom du Job TPT (Teradata Parallel Transporter) défini via la commande Teradata `tbuild`.  
|          | Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée. |

| Layout Name(schema) | Spécifiez un schéma pour le chargement des données.  
|                    | Ce champ apparaît uniquement lorsque la case **Set Script Parameters** est cochée. |

| Return mload error | Cochez cette case pour spécifier le code de retour à partir duquel retourner une erreur dans la console. |

| Define character set | Cochez cette case pour spécifier l'encodage des caractères à utiliser dans votre système. |

| Apply TPT consumer operator optional attributes | Cochez cette case afin de définir les attributs facultatifs pour l'opérateur du consommateur sélectionné.  

| Optional attributes | Cliquez sur le bouton `[+]` sous la table pour ajouter autant des lignes que nécessaire pour chaque attribut facultatif. Configurez les paramètres comme suit pour chaque attribut :  
|                    | • **Name** : cliquez dans la cellule et sélectionnez un attribut facultatif dans la liste déroulante. La liste des attributs facultatifs varie selon l'opérateur du consommateur sélectionné. Pour plus d’informations concernant les attributs facultatifs pour chaque opérateur de consommateur, consultez **Attributs facultatifs supportés pour chaque opérateur de consommateur** à la page 4151.  
|                    | • **Value** : saisissez la valeur de l’attribut facultatif.  
Cette table apparaît uniquement lorsque la case **Apply TPT consumer operator optional attributes** est cochée.

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

<table>
<thead>
<tr>
<th>EXIT_VALUE</th>
<th>code de sortie du processus. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td>EXIT_VALUE</td>
<td>code de sortie du processus. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable <strong>After</strong> et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case <strong>Die on error</strong> est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Il est nécessaire d’installer l’outil de client Teradata avec les fonctionnalités Teradata Parallel Transporter Base et Teradata Parallel Transporter Stream sur la machine où les Jobs utilisant ce composant sont exécutés.</td>
</tr>
</tbody>
</table>

### Attributs facultatifs supportés pour chaque opérateur de consommateur

Cette section liste tous les attributs facultatifs supportés pour chaque opérateur de consommateur.

### L’opérateur Load

BufferSize, ErrorLimit, MaxSessions, MinSessions, TenacityHours, TenacitySleep, AccountId, DataEncryption, DateForm, ErrorTable1, ErrorTable2, LogonMech, LogonMechData, NotifyExit,
Scénario : Charger des données dans une base de données Teradata

Ce scénario décrit un Job créant une table dans une base de données Teradata, écrivant des données dans un fichier délimité, chargeant ces données du fichier dans la table, récupérant les données de la table et les affichant dans la console.
**Construire le Job**

**Procédure**

1. Créez un Job et ajoutez les composants suivants en saisissant leur nom dans l’espace de modélisation graphique ou bien en les glissant-déposant de la Palette dans l’espace de modélisation graphique : un `tTeradataRow`, un `tFixedFlowInput`, un `tFileOutputDelimited`, un `tTeradataTPTExec`, un `tTeradataInput` et un `tLogRow`.

2. Reliez le `tFixedFlowInput` au `tFileOutputDelimited` à l’aide d’un lien `Row > Main`.

3. Connectez le `tTeradataInput` au `tLogRow` avec le même type de lien.

4. Reliez le `tTeradataRow` au `tFixedFlowInput` à l’aide d’un lien `Trigger > OnSubjobOk`.

5. Connectez le `tFixedFlowInput` au `tTeradataTPTExec` et le `tTeradataTPTExec` au `tTeradataInput` à l’aide du même type de lien.

**Configurer les composants**

**Créer une nouvelle table d’une base de données Teradata**

**Procédure**

1. Double-cliquez sur le `tTeradataRow` pour ouvrir sa vue `Basic settings`. 
2. Renseignez les champs **Host**, **Database**, **Username** et **Password** en saisissant respectivement le nom de l’hôte, le nom de la base de données, l’identifiant et le mot de passe de l’utilisateur, pour la connexion.

3. Dans le champ **Query**, saisissez l’instruction SQL suivante afin de créer une table nommée *person* contenant ces trois colonnes : *id, name, sex*.

```sql
CREATE SET TABLE samples.person,
FALLECK,
NO BEFORE JOURNAL,
NO AFTER JOURNAL
(
    id INTEGER NOT NULL,
    name VARCHAR(50),
    sex VARCHAR(20)
)
UNIQUE PRIMARY INDEX (id)
```

**Préparer les données source**

**Procédure**

1. Double-cliquez sur le composant **tFixedFlowInput** pour ouvrir sa vue **Basic settings**.
2. Cliquez sur le bouton [...] à côté du champ **Edit schema** pour ouvrir l'éditeur du schéma.

3. Cliquez trois fois sur le bouton [+] pour ajouter trois colonnes : *id* de **Type Integer**, *name* et *sex*, de **Type String**.

4. Cliquez sur **OK** pour fermer l'éditeur et acceptez la propagation lorsqu'une fenêtre vous propose de propager le schéma.

5. Dans la zone **Mode**, sélectionnez **Use Inline Content (delimited file)** et saisissez les données d'entrée dans le champ **Content**.

   1;Ford;Male
   2;Rose;Female
   3;Sabrina;Female
   4;Teddy;Male
   5;Kate;Male

6. Double-cliquez sur le **tFileOutputDelimited** pour ouvrir sa vue **Basic settings**.
Dans le champ **File Name**, spécifiez le fichier dans lequel écrire les données d'entrée. Dans cet exemple, saisissez *E:/person.csv*.

**Charger les données sources dans une table vide**

**Procédure**

1. Double-cliquez sur le *tTeradataTPTExec* pour ouvrir sa vue **Basic settings**.

2. Dans les champs **TDPID**, **Database name**, **Username** et **Password**, saisissez vos informations de connexion à la base de données Teradata.

3. Dans le champ **Table**, saisissez le nom de la table dans laquelle les données source seront chargées. Dans cet exemple, saisissez *person*.

4. Dans le champ **Script generated folder**, parcourez votre système jusqu'au répertoire sous lequel créer le fichier du script Teradata Parallel Transporter durant l'exécution du Job. Dans cet exemple, le chemin est *E:/*.

5. Dans le champ **Load file**, parcourez votre système jusqu'au fichier contenant les données source. Dans cet exemple, le chemin est *E:/person.csv*.

7. Cliquez sur le bouton `[...]` à côté du champ **Edit schema** to open the schema editor.

![Schema of tTeradataTPTExec_1](image)

Cliquez trois fois sur le bouton `+` pour ajouter trois colonnes : `id` de **Type Integer**, `name` et `sex`, de **Type String**. Le nom des colonnes `id` et `name` dans la colonne **Db Column** sont entourés par ", car ce sont des mots-clés de la base de données Teradata.

Cliquez sur **OK** pour valider ces modifications et fermer l’éditeur du schéma.

8. Cliquez sur l’onglet **Advanced settings** pour ouvrir sa vue. Cochez la case **Apply TPT consumer operator optional attributes** et cliquez quatre fois sur le bouton `+` sous la table **Optional attributes** pour ajouter les attributs nécessaires suivants : `ErrorLimit`, `ErrorTable1`, `QueryBandSessionInfo` et `TraceLevel`.

![tTeradataTPTExec_1](image)

**Remarque :**

Pour les attributs de type VARCHAR, saisissez les valeurs entre guillemets doubles.
Récupérez les données de la table de base de données Teradata

Procédure

1. Double-cliquez sur le tTeradataInput pour ouvrir sa vue Basic settings.

   ![tTeradataInput Configuration](image)

2. Dans le champ Table Name, saisissez le nom de la table de laquelle lire les données. Dans cet exemple, saisissez **person**.

3. Dans le champ Query, saisissez l'instruction SQL suivante pour récupérer les données de la table **person**.

   ```sql
   SELECT * FROM samples.person ORDER BY id
   ```


   ![Schema Editor](image)

5. Cliquez trois fois sur le bouton [+] pour ajouter trois colonnes : *id* de Type Integer, *name* et *sex*, de Type String. Le nom des colonnes *id* et *name* dans la colonne Db Column sont entourés par `\`, car ce sont des mots-clés de la base de données Teradata.

6. Cliquez sur OK pour fermer l’éditeur de schéma et acceptez la propagation proposée par la fenêtre qui s’ouvre.

7. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings.
Dans la zone **Mode**, sélectionnez l’option **Table (print values in cells of a table)** pour un affichage optimal des résultats.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour l’exécuter.

```plaintext
Starting job tTeradataTPTExec_Demo at 10:35 28/09/2014.
[statistics] connecting to socket on port 3553
[statistics] connected
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

[statistics] disconnected
Job tTeradataTPTExec_Demo ended at 10:35 28/09/2014. [exit code=0]
```

Les données écrites dans la table de la base de données Teradata spécifiée sont affichées dans la console.
tTeradataTPTUtility

Ce composant écrit les données entrantes dans un fichier puis de charger les données du fichier dans une base de données Teradata.

Le tTeradataTPTUtility combine les utilitaires des composants tTeradataFastLoad, tTeradataMultiLoad, tTeradataTPump et tTeradataFastExport et permet d’écrire des données entrantes dans un fichier puis de charger les données du fichier dans une base de données Teradata.

Propriétés du tTeradataTPTUtility Standard

Ces propriétés sont utilisées pour configurer le tTeradataTPTUtility s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataTPTUtility Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Basic settings Data file

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Peut être Built-In ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Le champ <strong>Filename</strong> est alors pré-rempli à l’aide des données collectées.</td>
</tr>
<tr>
<td>Filename</td>
<td>Spécifiez le fichier dans lequel sauvegarder vos données de sortie.</td>
</tr>
<tr>
<td>Append</td>
<td>Cochez cette case pour ajouter les données entrantes dans le fichier défini dans le champ <strong>Filename</strong>.</td>
</tr>
<tr>
<td>Property Type</td>
<td>Peut-être Built-In ou Repository.</td>
</tr>
<tr>
<td></td>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants concernant la connexion à la base de données sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Execution platform</td>
<td>Sélectionnez le type de système d'exploitation utilisé pour exécuter le Job, <strong>Windows</strong> ou <strong>Unix</strong>.</td>
</tr>
<tr>
<td>TDPID</td>
<td>Spécifiez le Director program identifier de Teradata. Cela peut-être le nom ou l’adresse IP de la base de données Teradata à laquelle vous accédez.</td>
</tr>
<tr>
<td>Database name</td>
<td>Saisissez dans ce champ le nom de la base de données Teradata.</td>
</tr>
</tbody>
</table>
### Username et Password

Spécifiez l’identifiant et mot de passe de l’utilisateur de la base de données Teradata.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

### Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

**Remarque :**

Il est recommandé de ne pas utiliser le mot-clé Teradata Database comme nom de colonne de base de données. Si vous y êtes obligé, le nom de la colonne de base de données doit alors être entouré par ". Par exemple, lorsque le mot-clé id est utilisé comme nom de colonne d’une base de données, le champ Db Column doit être renseigné par "id".

### Built-In

Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

### Repository

Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Consumer Operator

Sélectionnez un opérateur dans la liste déroulante.

- **Load** : Charge un grand volume de données dans une table vide dans la base de données Teradata à l’aide de l’utilitaire Teradata FastLoad utility.
- **Inserter** : Insère les données dans une table définie dans la base de données Teradata.
- **Update** : Réalise des insertions, des mises à jour ou des suppressions dans une table nouvellement créée ou déjà existante de la base de données Teradata à l'aide de l'utilitaire Teradata MultiLoad utility.

- **Stream** : Réalise des insertions, des mises à jour ou des suppressions dans une table nouvellement créée ou déjà existante de la base de données Teradata à l'aide de l'utilitaire Teradata TPump.


### Action on data

Sélectionnez dans la liste déroulante une action à effectuer sur les données.

- **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.

- **Update** : Mettre à jour des entrées déjà existantes dans la table.

- **InsertOrUpdate** : Ajouter de nouvelles entrées. Une mise à jour sera effectuée si l'entrée avec la référence donnée existe déjà.

- **Delete** : Supprimer des entrées correspondant au flux d'entrée.

**Remarque :**

Vous devez spécifier au moins une colonne comme clé primaire sur laquelle baser l'opération **Update** ou **Delete**. Cela est faisable en cliquant sur le bouton [...] à côté du champ **Edit schema** et en cochant la case de la (des) colonne(s) à définir comme clé(s) primaire(s).

Ce champ apparaît uniquement lorsque l'opérateur **Update** ou **Stream** est sélectionné dans la liste déroulante **Consumer Operator**.

### Producer Operator

Sélectionnez un opérateur dans la liste déroulante. Seul l'opérateur **DataConnector** est actuellement supporté.

**DataConnector** : accède aux fichiers directement ou via un module d'accès, puis les écrit dans le flux de données.


### Table

Spécifiez le nom de la table à écrire dans la base de données Teradata. Notez que vous ne pouvez écrire qu’une table à la fois.
### Advanced settings

<table>
<thead>
<tr>
<th><strong>Script generated folder</strong></th>
<th>Spécifiez le répertoire sous lequel le fichier script Teradata Parallel Transporter sera créé durant l’exécution du Job puis supprimé avant la fin du Job.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Error file</strong></td>
<td>Spécifiez le fichier dans lequel les messages de log seront enregistrés.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Row separator</strong></th>
<th>Caractère, chaîne ou expression régulière séparant les lignes.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field separator</strong></td>
<td>Caractère, chaîne ou expression régulière séparant les champs.</td>
</tr>
<tr>
<td><strong>Include header</strong></td>
<td>Cochez cette case pour inclure l’en-tête des colonnes dans le fichier.</td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
<td>Sélectionnez l’encodage à partir de la liste ou sélectionnez Custom et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.</td>
</tr>
<tr>
<td><strong>Set Script Parameters</strong></td>
<td>Cochez cette case pour spécifier la valeur des paramètres Load Operator, Data Connector, Job Name, et Layout Name (schema) qui seront utilisés lors de la génération d’un script durant l’exécution du Job. Si vous ne les spécifiez pas manuellement, le système utilisera leurs valeurs par défaut.</td>
</tr>
<tr>
<td><strong>Load Operator</strong></td>
<td>Spécifiez l’opérateur de chargement. Ce champ apparaît uniquement lorsque la case Set Script Parameters est cochée.</td>
</tr>
<tr>
<td><strong>Data Connector</strong></td>
<td>Spécifiez le connecteur de données. Ce champ apparaît uniquement lorsque la case Set Script Parameters est cochée.</td>
</tr>
<tr>
<td><strong>Layout Name(schema)</strong></td>
<td>Spécifiez un schéma pour le chargement des données. Ce champ apparaît uniquement lorsque la case Set Script Parameters est cochée.</td>
</tr>
<tr>
<td><strong>Define Log table</strong></td>
<td>Cochez cette case pour spécifiez le nom de table de log.</td>
</tr>
</tbody>
</table>
### Return mload error
Cochez cette case pour spécifier le code de retour à partir duquel retourner une erreur dans la console.

### Define character set
Cochez cette case pour spécifier l'encodage des caractères à utiliser dans votre système.

### Apply TPT consumer operator optional attributes
Cochez cette case afin de définir les attributs facultatifs pour l’opérateur du consommateur sélectionné.

### Optional attributes
Cliquez sur le bouton [+] sous la table pour ajouter autant des lignes que nécessaire pour chaque attribut facultatif. Configurez les paramètres comme suit pour chaque attribut :

- **Name** : cliquez dans la cellule et sélectionnez un attribut facultatif dans la liste déroulante. La liste des attributs facultatifs varie selon l’opérateur du consommateur sélectionné. Pour plus d’informations concernant les attributs facultatifs pour chaque opérateur du consommateur, consultez Attributs facultatifs supportés pour chaque opérateur de consommateur à la page 4151.

- **Value** : saisissez la valeur de l’attribut facultatif.


Cette table apparaît uniquement lorsque la case **Apply TPT consumer operator optional attributes** est cochée.

### tStatCatcher Statistics
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>EXIT_VALUE : code de sortie du processus. Cette variable est une variable After et retourne un nombre entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td></td>
<td>Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.</td>
</tr>
<tr>
<td></td>
<td>Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace</td>
</tr>
</tbody>
</table>


pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Il est nécessaire d’installer l’outil de client Teradata avec les fonctionnalités Teradata Parallel Transporter Base et Teradata Parallel Transporter Stream sur la machine où les Jobs utilisant ce composant sont exécutés.</td>
</tr>
</tbody>
</table>

**Scénario associé**

Pour un scénario associé, consultez Scénario : Charger des données dans une base de données Teradata à la page 4152.
**tTeradataTPump**

Ce composant insère, met à jour ou supprime des données dans la base de données Teradata, à l’aide de l’utilitaire de chargement TPump, qui permet d’obtenir des données en quasi temps réel dans un entrepôt de données.

Utilisez ce composant particulièrement dans les environnements où les fenêtres de lot sont réduites et où la maintenance des entrepôts coïncide avec les heures normales de travail.

**Propriétés du tTeradataTPump Standard**

Ces propriétés sont utilisées pour configurer le tTeradataTPump s’exécutant dans le framework de Jobs Standard.

Le composant tTeradataTPump Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property type</strong></td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td><strong>Execution platform</strong></td>
<td>Sélectionnez le type de système d’exploitation que vous utilisez.</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>Nom de l’hôte ou l’adresse IP du serveur de la base de données.</td>
</tr>
<tr>
<td><strong>Database name</strong></td>
<td>Nom de la base de données.</td>
</tr>
</tbody>
</table>
| **Username et Password** | Informations d’authentification de l’utilisateur de base de données.  
Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Table**             | Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois. |
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. |
**Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.


Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

**Script generated folder**

Parcourez votre répertoire et sélectionnez la destination du fichier qui sera créé.

**Action to data**

Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :

- **Insert** : Ajouter de nouvelles entrées à la table. Le Job s'arrête lorsqu'il détecte des doublons.
- **Update** : Mettre à jour les entrées existantes.
- **Insert or update** : insère un nouvel enregistrement. Si l'enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- **Delete** : Supprimer les entrées correspondantes au flux d’entrée.

**Avertissement** :

Il est nécessaire de spécifier au minimum une clé de recherche sur laquelle baser les opérations **Update** et **Delete**. Il est possible de définir les colonnes qui agiront comme clé de recherche à partir du schéma, pour une utilisation de base, ou à partir des options avancées (**Advanced settings**) pour une utilisation optimisée de ces opérations.

**Where condition in case Delete**

Saisissez une condition, qui, si elle est vérifiée, supprimera la ligne.

Ce champ apparaît uniquement lorsque **Delete** est sélectionné dans la liste déroulante **Action to data**.
| **Load file** | Entrez le nom du fichier à partir duquel vous souhaitez charger les données. |
| **Field separator** | Caractère, chaîne ou expression régulière séparant les champs. |
| **Error file** | Parcourez votre répertoire et sélectionnez la destination du fichier dans lesquels seront enregistrés les messages d’erreur. |

**Advanced settings**

| **Define Log table** | Cochez cette case pour définir la table de log que vous souhaitez utiliser au lieu de celle par défaut, à savoir celle que vous avez définie dans l’onglet Basic settings. La syntaxe requise pour définir la table de log est databasename.logtablename. |
| **BEGIN LOAD** | Ce champ vous permet de définir votre commande BEGIN LOAD pour lancer ou relancer une tâche TPump. Vous pouvez spécifier le nombre de sessions à utiliser, la limite d’erreurs, ainsi que tout autre paramètre nécessaire à l’exécution de la tâche. La valeur par défaut est : SESSIONS 8 PACK 600 ARRAYSUPPORT ON CHECKPOINT 60 TENACITY 2 ERRLIMIT 1000. Pour plus d’informations, consultez la documentation Teradata Parallel Data Pump Reference. |
| **Return tpump error** | Cochez cette case pour spécifier le code de retour à partir duquel retourner une erreur dans la console. |
| **Define character set** | Spécifiez l’encodage des caractères dont vous avez besoin pour utiliser votre système. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| **Global Variables** | **EXIT_VALUE** : code de sortie du processus. Cette variable est une variable After et retourne un nombre entier. **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette |
Règle d’utilisation | Ce composant couvre toutes les possibilités de requête SQL dans les bases de données Teradata.

**Scénario : Insérer des données dans une table d’une base de données Teradata**

Dans ce scénario, l’objectif est de créer un Job utilisant le composant **tTeradataTPump** pour insérer des données clients dans une table d’une base de données Teradata et retourner la valeur de retour lorsqu’une erreur se produit.

Trois composants sont utilisés pour implémenter cette tâche :

- **tRowGenerator** : génère autant de lignes que nécessaire, en utilisant au hasard des données clients prises dans une liste.
- **tFileOutputDelimited** : écrit les données clients dans un fichier délimité.
- **tTeradataTPump** : insère les données clients dans la table de la base de données Teradata en mode Tpump.

**Déposer les composants**

**Procédure**

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : **tRowGenerator, tFileOutputDelimited, tTeradataTPump**.
2. Liez le **tRowGenerator** au **tFileOutputDelimited** à l’aide d’un lien **Row > Main**.
3. Liez le **tRowGenerator** au **tTeradataTPump** à l’aide d’un lien **Trigger > OnSubjobOk**.
Configurer les composants

Procédure

1. Double-cliquez sur le tRowGenerator pour ouvrir la fenêtre [RowGenerator Editor].
   Dans la fenêtre [RowGenerator Editor], définissez les données à générer. Pour ce Job, le schéma se compose de deux colonnes : ID et Name.

<table>
<thead>
<tr>
<th>Column</th>
<th>Key</th>
<th>Type</th>
<th>N.</th>
<th>Length</th>
<th>Precision</th>
<th>Default</th>
<th>Comment</th>
<th>Functions</th>
<th>Environ.</th>
<th>Preview</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>✓</td>
<td>int</td>
<td>✓</td>
<td>10</td>
<td>0</td>
<td></td>
<td></td>
<td>sequence</td>
<td>sequence</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>✓</td>
<td>String</td>
<td>✓</td>
<td>20</td>
<td>0</td>
<td></td>
<td></td>
<td>GetLast...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dans le champ **Number of Rows for RowGenerator**, saisissez le nombre de lignes à générer.

2. Double-cliquez sur le tFileOutputDelimited afin d’afficher sa vue Component et définir ses propriétés.

3. À côté du champ **File Name**, parcourez votre répertoire jusqu’au fichier de sortie, ou saisissez un nom pour le fichier à créer.

4. Entre guillemets doubles, saisissez les séparateurs à utiliser, à côté des champs **Row Separator** et **Field Separator**, par exemple \n et un point-virgule.
5. Double-cliquez sur le composant `tTeradataTPump` afin d'ouvrir sa vue `Component`.

Dans l'onglet `Basic settings` de la vue `Component`, définissez les paramètres du composant `tTeradataTPump`.

6. Saisissez le nom de la base de données dans le champ `Database name`, votre identifiant de connexion à la base de données dans le champ `User name`, et votre mot de passe dans le champ `Password`.

7. Spécifiez la table dans laquelle insérer les données clients. Dans ce scénario, la table est `mytable`.

8. Dans les champs `Script generated folder`, `Load file` et `Error file`, spécifiez le répertoire respectivement au dossier dans lequel vous stockez les fichiers de scripts générés.

9. Dans le champ `Load File`, spécifiez le fichier contenant les données clients à insérer.

10. Dans le champ `Error file`, spécifiez le fichier contenant les informations concernant les erreurs.

11. Dans le champ `Action on data`, sélectionnez l'option `Insert`.

Cliquez sur `Edit schema` et vérifiez que le schéma est retrouvé à partir du schéma d'entrée. Au besoin, cliquez sur `Sync Columns`.
**Exécuter le Job**

**Procédure**

1. Appuyez sur **F6** pour exécuter le Job.
2. La console de la vue **Run** s'affiche ainsi :

```
Running job 'tpump'...
Starting job tpump at 17:01 19/07/2010.
[statistics] connecting to socket on port 3740
[statistics] connected
[statistics] disconnected
Job tpump ended at 17:02 19/07/2010 [exit code=0]
```

3. Double-cliquez sur le composant **tTeradataTPump** pour retourner à sa vue **Component**.
4. Dans l'onglet **Advanced settings**, cochez la case **Return tpump error** et saisissez le code de retour à partir duquel retourner une erreur dans la console. Dans cet exemple, saisissez le chiffre 4 et utilisez les valeurs par défaut pour les autres paramètres.
5. Appuyez sur **F6** pour exécuter le Job.
6. La console de la vue **Run** s'affiche ainsi :

```
Running job 'tpump'...
Starting job tpump at 17:56 19/07/2010.
[statistics] connecting to socket on port 3549
[statistics] connected
[statistics] disconnected
Exception in component tTeradataTPump_1
java.lang.RuntimeException: TPump returned exit code 12
at bug.tpump_0_1.tpump.TTeradataTPump_1Process(tpump.java:307)
at bug.tpump_0_1.tpump.runJobInTOS(tpump.java:504)
at bug.tpump_0_1.tpump.main(tpump.java:378)
Job tpump ended at 17:56 19/07/2010. [exit code=0]
```

**Résultats**

Une erreur se produit et le **TPump** retourne le code de retour, de 12. Si vous avez besoin d'informations détaillées concernant les informations détaillées de l'erreur, vous pouvez ouvrir le fichier de log stocké dans le répertoire spécifié dans le champ **Error file** et dans l'onglet **Basic settings** de la vue **Component**.
**tUniqRow**

Ce composant assure une qualité de données des flux d’entrée et de sortie du Job.
Le composant tUniqRow compare les entrées et supprime les doublons du flux d’entrée.

**Propriétés du tUniqRow Standard**

Ces propriétés sont utilisées pour configurer le tUniqRow s’exécutant dans le framework de Jobs Standard.
Le composant tUniqRow Standard appartient à la famille Data Quality.
Le composant de ce framework est toujours disponible.

**Basic settings**

| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
- View schema : sélectionnez cette option afin de voir le schéma.
- Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

| Built-In | Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend. |
| Repository | Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend. |
| Unique key | Sélectionnez dans cette zone une ou plusieurs colonnes sur lesquelles le dédoublonnage sera effectué.
- Cochez la case Key attribute afin d’effectuer le dédoublonnage sur toutes les colonnes. |
- Cochez la case **Case sensitive** afin de différencier les majuscules et les minuscules.

### Advanced settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Only once each duplicated key</strong></td>
<td>Cochez cette case si vous souhaitez envoyer uniquement les premières entrées en doublon des colonnes définies comme clé(s) vers le flux de sortie des doublons.</td>
</tr>
</tbody>
</table>
| **Use of disk (suitable for processing large row set)** | Cochez cette case pour permettre la génération de fichiers temporaires sur le disque dur lors du traitement de données volumineuses. Cela permet d’empêcher l’échec de l’exécution d’un Job dû à un débordement de la mémoire. Quand cette case est cochée, vous devez également définir les éléments suivants :  
  - **Buffer size in memory**: Sélectionnez le nombre de lignes à mettre en mémoire tampon avant qu’un fichier temporaire ne soit généré sur le disque dur.  
  - **Directory for temp files**: Indiquez l’endroit où les fichiers temporaires doivent être enregistrés.  

  **Avertissement** :  
  Assurez-vous que le répertoire indiqué pour vos fichiers temporaires existe, sans quoi l’exécution du Job échouera.  

| **Ignore trailing zeros for BigDecimal**     | Cochez cette case pour ignorer les zéros en fin de champs pour les données de type BigDecimal. |
| **tStatCatcher Statistics**                  | Cochez cette case pour collecter les métadonnées de process du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_UNIQUES</strong></td>
<td>nombre de lignes uniques. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
<tr>
<td><strong>NB_DUPLICATES</strong></td>
<td>nombre de lignes en doublon. Cette variable est une variable <strong>After</strong> et retourne un nombre entier.</td>
</tr>
</tbody>
</table>
| **ERROR_MESSAGE**| message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  

Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.  

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

## Utilisation

### Règle d’utilisation

Ce composant est un composant intermédiaire puisqu’il sert à traiter un flux de données. Ainsi, il nécessite un composant d’entrée et un composant de sortie.

## Scénario 1 : Dédoublonner des données

Ce scénario décrit un Job à cinq composants, dont le but est de trier des entrées à partir d’une liste d’entrée comprenant des noms, de trouver des noms en doublons, et d’afficher les noms uniques ainsi que les noms en doublon dans la console Run.

### Construire le Job

**Procédure**

1. A partir de la Palette, cliquez et déposez les composants suivants dans l’espace de modélisation graphique : un tFileInputDelimited, un tSortRow, un tUniqRow, ainsi que deux composants tLogRow, et nommez-les comme illustré ci-dessus.
2. Connectez les composants tFileInputDelimited, tSortRow, et tUniqRow à l’aide de liens Row > Main.
3. Connectez le tUniqRow au premier tLogRow à l’aide d’un lien Main > Uniques.
4. Connectez le composant tUniqRow au second tLogRow via une connexion de type Main > Duplicates.

###Configurer les composants

**Procédure**

1. Double-cliquez sur le composant tFileInputDelimited afin d’afficher l’onglet Basic settings de sa vue Component.
2. Cliquez sur le bouton [...] à côté du champ File Name pour sélectionner le fichier contenant les données d’entrée.

3. Définissez l’en-tête (Header) et le pied de page (Footer). Dans ce scénario, la ligne d’en-tête est la première ligne du fichier d’entrée.

4. Cliquez sur Edit schema pour définir le schéma de ce composant. Dans ce scénario, le fichier d’entrée comprend cinq colonnes : Id, FirstName, LastName, Age, et City. Cliquez ensuite sur OK pour propager le schéma et fermer l’éditeur de schéma.

5. Double-cliquez sur le tSortRow afin d’afficher sa vue Basic settings.

6. Afin de disposer les entrées en fonction de l’ordre alphabétique des noms, ajoutez deux lignes au tableau Criteria en cliquant sur le bouton [+], sélectionnez les colonnes FirstName et LastName sous Schema column, sélectionnez le type de tri alphabétique (alpha), puis sélectionnez l’ordre ascendant (asc).

7. Double-cliquez sur le composant tUniqRow afin d’afficher l’onglet Basic settings de sa vue Component.
Dans la zone **Unique key**, sélectionnez les colonnes sur lesquelles vous souhaitez effectuer le dédoublonnage. Dans ce scénario, les noms en doublon seront triés.

Dans l'onglet **Basic settings** de la vue **Component** des deux composants **tLogRow**, cochez l'option **Table** afin de visualiser le résultat de l'exécution du Job en mode tableau.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Sauvegardez votre Job à l'aide des touches **Ctrl+S**.
2. Exécutez le Job en appuyant sur la touche **F6** ou en cliquant sur le bouton **Run** de l'onglet **Run**.

Dans la console **Run**, les noms uniques et les noms en doublon sont affichés dans des tableaux différents.
tUnite

Ce composant centralise des données provenant de sources diverses et hétérogènes.
Il fusionne des données de diverses sources, basées sur un même schéma.
Le tUnite ne peut exister dans une boucle de flux de données. Par exemple, si un flux de données passe par plusieurs composants tMap afin de générer deux flux, ils ne peuvent alimenter un tUnite.

Propriétés du tUnite Standard

Ces propriétés sont utilisées pour configurer le tUnite s’exécutant dans le framework de Jobs Standard.
Le composant tUnite Standard appartient à la famille Orchestration.
Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
|• View schema : sélectionnez cette option afin de voir le schéma.
|• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
|• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content.
|Cliquez sur Sync columns pour récupérer le schéma du composant précédent dans le Job. |
| Built-in | Le schéma est créé et conservé pour ce composant uniquement. Voir également le Guide utilisateur du Studio Talend |
### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option. |
| | NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier. |
| | Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant. |
| | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
| | Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

### Utilisation

| Règle d’utilisation | Ce composant n’est pas un composant de début; il requiert un ou plusieurs composants en entrée et un composant de sortie. |
| Connections | Liens de sortie (de ce composant à un autre) :
- **Row** : Main.
- **Trigger** : Run if, OnComponentOk, OnComponentError. |
| | Liens d’entrée (d’un autre composant à celui-ci) :
- **Row** : Main, Reject. |
| | Pour plus d’informations concernant les liens, consultez le Guide utilisateur du Studio Talend. |

### Scénario : Itération sur des fichiers et fusion de contenu

Le Job suivant effectue une itération sur une liste de fichiers puis fusionne leur contenu et affiche le contenu final des deux colonnes dans la console.
Déposer et relier les composants

Procédure

1. Cliquez et déposez les composants suivants dans l'espace de modélisation : tFileList, tFileInputDelimited, tUnite et tLogRow.
2. Connectez le composant tFileList au tFileInputDelimited à l'aide d'un lien Iterate et connectez les autres composants à l'aide de liens Row main.

Configurer les composants

Procédure

1. Dans l'onglet Basic settings du composant tFileList, sélectionnez le répertoire dans lequel les fichiers à fusionner sont stockés.

Les fichiers sont très basiques et contiennent une liste de pays et leur score respectif.

2. Dans la liste Case Sensitive, sélectionnez Yes afin de prendre la casse en compte.
3. Sélectionnez le composant **tFileInputDelimited** et affichez son onglet **Basic settings**.

4. Pour renseigner le champ **File Name/Steam**, appuyez sur **Ctrl+Espace** pour accéder à la liste des variables. Pour traiter tous les fichiers du répertoire défini dans le composant **tFileList**, sélectionnez **tFileList.CURRENT_FILEPATH** dans la liste des variables globales.

5. Cliquez sur le bouton **Edit Schema** et définissez manuellement le schéma à deux colonnes pour qu’il corresponde au contenu des fichiers d’entrée.

Pour cet exemple, les deux colonnes sont **Country** et **Points**. Elles peuvent toutes les deux prendre une valeur nulle, ainsi cochez les cases de la colonne** Nullable**. La colonne **Country** sera de type **String**, et la colonne **Points** sera de type **Integer**.

6. Cliquez sur **OK** pour valider les paramètres, puis acceptez la propagation du schéma dans le reste du Job.

7. Puis sélectionnez le composant **tuUnit** et affichez la vue **Component**. Notez que le schéma de sortie reflète exactement celui d’entrée et qu’il est en lecture seule.

8. Dans la vue **Component** du composant **tLogRow**, sélectionnez l’option **Table (Print values in cells of a table)** pour afficher correctement les valeurs de sortie.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Enregistrez le Job en appuyant sur les touches **Ctrl+S**.
   La console affiche les données des différents fichiers, fusionnées dans une seule table.
Starting job fetch at 16:11 08/03/2010

<table>
<thead>
<tr>
<th>tLogRow_1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>Points</td>
</tr>
<tr>
<td>france</td>
<td>12</td>
</tr>
<tr>
<td>usa</td>
<td>10</td>
</tr>
<tr>
<td>france</td>
<td>14</td>
</tr>
<tr>
<td>uk</td>
<td>13</td>
</tr>
<tr>
<td>usa</td>
<td>9</td>
</tr>
<tr>
<td>france</td>
<td>5</td>
</tr>
<tr>
<td>france</td>
<td>16</td>
</tr>
<tr>
<td>uk</td>
<td>13</td>
</tr>
</tbody>
</table>

Job fetch ended at 16:11 08/03/2010. [exit code=0]
tVectorWiseCommit

Ce composant utilise une connexion unique pour commiter en une seule fois une transaction globale au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

Le tVectorWiseCommit valide les données traitées dans un Job à partir d’une base de données connectée.

Propriétés du tVectorWiseCommit Standard

Ces propriétés sont utilisées pour configurer le tVectorWiseCommit s’exécutant dans le framework de Jobs Standard.

Le composant tVectorWiseCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>S’il y a plus d’une connexion dans le Job en cours, sélectionnez le composant tVectorWiseConnection dans la liste.</td>
</tr>
<tr>
<td>Close connection</td>
<td>Cette option est cochée par défaut. Elle permet de fermer la connexion à la base de données une fois le commit effectué. Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
<tr>
<td></td>
<td>Avertissement :</td>
</tr>
<tr>
<td></td>
<td>Si vous utilisez un lien de type Row &gt; Main pour relier le tVectorWiseCommit à votre Job, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection car la connexion sera fermée avant la fin du commit de votre première ligne.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

4183
## Utilisation

<table>
<thead>
<tr>
<th>Règle d'utilisation</th>
<th>Ce composant est généralement utilisé avec des composants VectorWise et notamment avec les composants tVectorWiseConnection et tVectorWiseRollback.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic settings</td>
<td>Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable. Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

## Scénario associé

Pour plus d’informations relatives au fonctionnement du composant tVectorWiseCommit, consultez Scénario : Insérer des données dans des tables mère/fille à la page 2620.
tVectorWiseConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Propriétés du tVectorWiseConnection Standard

Ces propriétés sont utilisées pour configurer le tVectorWiseConnection s’exécutant dans le framework de Jobs Standard.

Le composant tVectorWiseConnection Standard appartient aux familles Databases et ELT.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong> : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td><strong>Repository</strong> : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
<td></td>
</tr>
<tr>
<td>Server</td>
<td>Adresse IP du serveur de base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données.</td>
</tr>
<tr>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
<td></td>
</tr>
<tr>
<td>Use or register a shared DB Connection</td>
<td>Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ <strong>Shared DB Connection Name</strong> qui s’affiche, saisissez un nom</td>
</tr>
</tbody>
</table>
pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée avec un composant **tRunJob** ayant une de ces options activée fera échouer votre Job.

### Advanced settings

| **Auto Commit** | Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée. Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commitées de manière explicite à l’aide du composant correspondant de commit. Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

### Utilisation

| **Règle d’utilisation** | Ce composant est généralement utilisé avec des composants VectorWise, notamment les composants **tVectorWiseCommit** et **tVectorWiseRollback**. |
| **Limitation** | Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton **Install** dans l’onglet **Component**. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet **Modules** de la perspective **Integration** de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (https://help.talend.com). |
Scénario associé

Pour un scénario associé au composant tVectorWiseConnection, consultez tMysqlConnection à la page 2618.
tVectorWiseInput

Ce composant exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma.

Le tVectorWiseInput lit une base de données et en extrait des champs à l’aide de requêtes. Les champs récupérés sont ensuite transmis au composant suivant via une connexion de flux (Main).

**Propriétés du tVectorWiseInput Standard**

Ces propriétés sont utilisées pour configurer le tVectorWiseInput s’exécutant dans le framework de Jobs Standard.

Le composant tVectorWiseInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être <strong>Built-in</strong> ou <strong>Repository</strong></td>
</tr>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>
| ![Icone de connexion existante](image) | Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue **Basic settings** du composant.  
Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend. |
| **Use an existing connection** | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie. |

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par...
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue **Basic settings** du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th><strong>Server</strong></th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé **line** lors du nommage des champs. **Built-in** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**. **Repository** : Le schéma existe déjà et est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**. Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
  - **View schema** : sélectionnez cette option afin de voir le schéma.  
  - **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
  - **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job |
Table name

Saisissez le nom de la table.

Query type et Query

Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

Guess Query

Cliquez sur le bouton Guess Query pour générer la requête correspondant au schéma de votre table dans le champ Query.

Guess schema

Cliquez sur le bouton pour récupérer le schéma de la table.

Advanced settings

Trim all the String/Char columns

Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes contenant des chaînes de caractères.

Trim column

Dans la colonne Column, saisissez le nom de la colonne dans laquelle supprimer les espaces en début et en fin de champ dans les colonnes sélectionnées.

tStatCatcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

Global Variables

NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant couvre toutes les possibilités de requête SQL dans les bases de données VectorWise.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

Scénario associé

Pour un scénario associé au composant **tVectorWisInput**, consultez :

- **Scénario :** Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520.
tVectorWiseOutput

Ce composant exécute l’action définie sur la table et/ou sur les données d’une table, en fonction du flux entrant provenant du composant précédent.

Le tVectorWiseOutput écrit, met à jour, modifie ou supprime les données d’une base de données.

Propriétés du tVectorWiseOutput Standard

Ces propriétés sont utilisées pour configurer le tVectorWiseOutput s’exécutant dans le framework de Jobs Standard.

Le composant tVectorWiseOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
</tr>
</tbody>
</table>

Cliquez sur cette icône pour ouvrir l’assistant de configuration de connexion à la base de données et enregistrer les paramètres de connexion que vous avez définis dans la vue Basic settings du composant.

Pour plus d’informations sur comment définir et stocker des paramètres de connexion de base de données, consultez le Guide utilisateur du Studio Talend.

Use an existing connection

Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par
exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans la vue Basic settings du composant de connexion créant cette connexion.

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d'authentification de l'utilisateur de base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Table</td>
<td>Nom de la table à créer. Vous ne pouvez créer qu’une seule table à la fois.</td>
</tr>
<tr>
<td>Action on table</td>
<td>Vous pouvez effectuer l’une des opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td><strong>None</strong> : n’effectuer aucune opération de table.</td>
</tr>
<tr>
<td></td>
<td><strong>Drop and create the table</strong> : supprimer la table puis en créer une nouvelle.</td>
</tr>
<tr>
<td></td>
<td><strong>Create a table</strong> : créer une table qui n’existe pas encore.</td>
</tr>
<tr>
<td></td>
<td><strong>Create table if doesn’t exist</strong> : créer la table si nécessaire.</td>
</tr>
<tr>
<td></td>
<td><strong>Clear a table</strong> : supprimer le contenu de la table.</td>
</tr>
<tr>
<td>Action on data</td>
<td>Vous pouvez effectuer les opérations suivantes sur les données de la table sélectionnée :</td>
</tr>
<tr>
<td></td>
<td><strong>Insert</strong> : Ajouter de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.</td>
</tr>
<tr>
<td></td>
<td><strong>Update</strong> : Mettre à jour les entrées existantes.</td>
</tr>
<tr>
<td></td>
<td><strong>Insert or update</strong> : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.</td>
</tr>
<tr>
<td></td>
<td><strong>Update or insert</strong> : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.</td>
</tr>
</tbody>
</table>
**Delete** : Supprimer les entrées correspondantes au flux d’entrée.

⚠️ **Avertissement** :

Il est nécessaire de spécifier au minimum une colonne comme clé primaire sur laquelle baser les opérations **Update** et **Delete**. Pour cela, cliquez sur le bouton `[...]` à côté du champ **Edit Schema** et cochez la ou les case(s) correspondant à la ou aux colonne(s) que vous souhaitez définir comme clé(s) primaire(s). Pour une utilisation avancée, cliquez sur l’onglet **Advanced settings** pour définir simultanément les clés primaires sur lesquelles baser les opérations de mise à jour (Update) et de suppression (Delete). Pour cela, cochez la case **Use field options** et sélectionnez la case **Key in update** correspondant à la colonne sur laquelle baser votre opération de mise à jour (Update). Procédez de la même manière avec les cases **Key in delete** pour les opérations de suppression (Delete).

<table>
<thead>
<tr>
<th>Schema et <strong>Edit Schema</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé <code>line</code> lors du nommage des champs.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Built-In</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Repository</strong></th>
</tr>
</thead>
</table>
| Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.  
Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.  
Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center ([https://help.talend.com](https://help.talend.com)). |

<table>
<thead>
<tr>
<th>Cliquez sur <strong>Edit schema</strong> pour modifier le schéma. Si le schéma est en mode <strong>Repository</strong>, trois options sont disponibles :</th>
</tr>
</thead>
</table>
| • **View schema** : sélectionnez cette option afin de voir le schéma.  
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les |
modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

| **Die on error** | Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**. |

| **Advanced settings** |  |
| **Commit every** | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution. |
| **Additional Columns** | Cette option n’est pas disponible si vous venez de créer la table de données (que vous l’ayez préalablement supprimée ou non). Cette option vous permet d’effectuer des actions sur les colonnes, à l’exclusion des actions d’insertion, de mise à jour, de suppression ou qui nécessitent un prétraitement particulier. |
| **Name** | Saisissez le nom de la colonne à modifier ou à insérer. |
| **SQL expression** | Saisissez la déclaration SQL à exécuter pour modifier ou insérer les données dans les colonnes correspondantes. |
| **Position** | Sélectionnez **Before**, **Replace** ou **After**, en fonction de l’action à effectuer sur la colonne de référence. |
| **Reference column** | Saisissez une colonne de référence que le composant **tTeradataOutput** peut utiliser pour situer ou remplacer la nouvelle colonne ou celle à modifier. |
| **Use field options** | Cochez cette case pour personnaliser une requête, surtout lorsqu’il y a plusieurs actions sur les données. |
| **Enable debug mode** | Cochez cette case pour afficher chaque étape du processus de d’écriture dans la base de données. |
| **Support null in “SQL WHERE” statement** | Cochez cette case pour prendre en compte les valeurs Null d’une table de base de données. |
| **Remarque** | Assurez-vous que la case **Nullable** est bien cochée pour les colonnes du schéma correspondantes. |
Use Batch

Cochez cette case pour activer le mode de traitement par lots pour le traitement des données.
Cette option n’est supportée que sur la version 2.5 et plus de Vectorwise, et n’est disponible que si vous sélectionnez Insert dans la liste déroulante Action on data dans l’onglet Basic settings.

Batch Size

Spécifiez le nombre d’enregistrements à traiter dans chaque lot.
Ce champ est disponible uniquement lorsque la case Use batch mode est cochée.

tStat Catcher Statistics

Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_UPDATED</td>
<td>nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_INSERTED</td>
<td>nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_DELETED</td>
<td>nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_REJECTED</td>
<td>nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>

Utilisation

Règle d’utilisation

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL. Il permet de faire des actions sur une table ou les données d’une table d’une base de données VectorWise. Il permet aussi de créer un flux de rejet avec...

**Scénario associé**

Pour un scénario associé, consultez :

- *Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput* à la page 2667 du composant tMysqlOutput.
tVectorWiseRollback

Ce composant annule la transaction commitée dans la base de données VectorWise connectée.

Propriétés du tVectorWiseRollback Standard

Ces propriétés sont utilisées pour configurer le tVectorWiseRollback s’exécutant dans le framework de Jobs Standard.
Le composant tVectorWiseRollback Standard appartient à la famille Databases.
Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component list</td>
<td>Sélectionnez le composant tVectorWiseConnection dans la liste s’il y a plus d’une connexion dans votre Job.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Décochez cette case pour continuer à utiliser la connexion une fois que le composant a exécuté sa tâche.</td>
</tr>
</tbody>
</table>

Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé avec d’autres composants VectorWise, notamment les composants tVectorWiseConnection et tVectorWiseCommit. |
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. |
| La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre |
Pour des exemples relatifs à l'utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l'aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l'aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d'informations concernant les paramètres dynamiques et les variables de contexte, consultez le Guide utilisateur du Studio Talend.

Scénario associé

Pour un scénario associé au composant tVectorWiseRollback, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623.
tVectorWiseRow

Ce composant agit sur la structure même de la base de données ou sur les données (mais sans les manipuler).

Le tVectorWiseRow est le composant spécifique à ce type de base de données. Il exécute des requêtes SQL déclarées sur la base de données spécifiée. Le suffixe Row signifie que le composant met en place un flux dans le Job bien que ce composant ne produise pas de données en sortie.

Le SQLBuilder peut vous aider à rapidement et aisément écrire vos requêtes.

Propriétés du tVectorWiseRow Standard

Ces propriétés sont utilisées pour configurer le tVectorWiseRow s’exécutant dans le framework de Jobs Standard.

Le composant tVectorWiseRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property type</td>
<td>Peut être Built-in ou Repository</td>
</tr>
<tr>
<td>Built-in</td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td>Repository</td>
<td>Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
</tbody>
</table>

Remarque :

Lorsqu’un Job contient un Job parent et un Job enfant, si vous souhaitez utiliser une connexion existant entre les deux niveaux, par exemple pour partager la connexion créée par le Job parent avec le Job enfant, vous devez :

1. Au niveau parent, enregistrer la connexion à la base de données à partager, dans
<table>
<thead>
<tr>
<th>Host</th>
<th>Adresse IP du serveur de base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Numéro du port d’écoute du serveur de base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Nom de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Informations d’authentification de l’utilisateur de base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :  
• View schema : sélectionnez cette option afin de voir le schéma.  
• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.  
• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |

2. Au niveau enfant, utiliser un composant dédié à la création de connexion, afin de lire la connexion enregistrée.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.
### Table Name
Nom de la table à traiter.

### Query type
La requête peut être **Built-in** ou distante dans le **Repository**

**Built-in** : Saisissez manuellement votre requête ou construisez-la à l’aide de SQLBuilder.

**Repository** : Sélectionnez la requête appropriée dans le Repository. Le champ **Query** est renseigné automatiquement.

### Guess Query
Cliquez sur le bouton **Guess Query** pour générer la requête correspondant au schéma de votre table dans le champ **Query**.

### Query
Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.

### Die on error
Cette case est cochée par défaut et stoppe le Job en cas d’erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreur, et ignorer les lignes en erreur. Vous pouvez récupérer les lignes en erreur, si vous le souhaitez. Pour cela, utilisez un lien **Row > Rejects**.

### Advanced settings

#### Propagate QUERY’s recordset
Cochez cette case pour insérer les résultats de la requête dans une colonne du flux en cours. Sélectionnez cette colonne dans la liste **use column**.

**Remarque** :
Cette option permet au composant d’avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type **Object**. Ce composant est généralement suivi du **tParseRecordSet**.

#### Use PreparedStatement
Cochez cette case pour utiliser une instance **PreparedStatement** afin de requêter votre base de données. Dans le tableau **Set PreparedStatement Parameter**, définissez les valeurs des paramètres représentés par des “?” dans l’instruction SQL définie dans le champ **Query** de l’onglet **Basic settings**.

- **Parameter Index** : Saisissez la position du paramètre dans l’instruction SQL.
- **Parameter Type** : Saisissez le type du paramètre.
- **Parameter Value** : Saisissez la valeur du paramètre.

**Remarque** :
Cette option est très utile si vous devez effectuer de nombreuses fois la même requête. Elle permet un gain de performance.
Commit every | Nombre de lignes à inclure dans le lot avant de commencer l’écriture dans la base. Cette option garantit la qualité de la transaction (cependant pas de rollback) et surtout une meilleure performance d’exécution.

tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant.

Global Variables

Global Variables

QUERY : requête traitée. Cette variable est une variable Flow et retourne une chaîne de caractères.
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.

Limitation


Scénario associé

Pour un scénario associé, consultez :
- Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
- Procédure du composant tDBSQLRow.
• **Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du composant tMysqlRow.**
tVerticaBulkExec

Ce composant vous permet de charger les données dans une table d’une base de données Vertica à partir d’un fichier local à l’aide de l’instruction Vertica COPY SQL. Pour plus d’informations concernant l’instruction Vertica COPY SQL, consultez COPY (en anglais).

Les composants tVerticaOutputBulk et tVerticaBulkExec sont généralement utilisés ensemble comme deux parties d’un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l’opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tVerticaOutputBulkExec. L’avantage d’utiliser deux composants séparés réside dans le fait de pouvoir transformer les données avant leur chargement dans la base de données.

Propriétés du tVerticaBulkExec Standard

Ces propriétés sont utilisées pour configurer le tVerticaBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tVerticaBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td>• Built-In : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
<td></td>
</tr>
<tr>
<td>• Repository : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
<td></td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version de la base de données.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong></td>
</tr>
</tbody>
</table>
pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, si vous devez partager une connexion existante entre ces deux niveaux, par exemple pour partager la connexion créée par le Job père au Job fils, vous devez :

1. au niveau du Job père, enregistrer la connexion à la base de données à partager dans la vue **Basic settings** du composant de connexion créant cette connexion à la base de données.

2. au niveau du Job fils, utiliser un composant de connexion dédié afin de lire cette connexion enregistrée.

Pour un exemple concernant le partage de connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

| **Host** | Saisissez l’adresse IP ou le nom d’hôte de la base de données. |
| **Port** | Saisissez le numéro du port d’écoute de la base de données. |
| **Database** | Saisissez le nom de la base de données. |
| **Schema** | Schéma de la base de données. |
| **Username et Password** | Saisissez les données d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres. |
| **Table** | Saisissez le nom de la table dans laquelle écrire les données. |
| **Action on table** | Sélectionnez une opération à effectuer sur la table définie, vous pouvez effectuer l'une des opérations suivantes :
- **Default** : aucune opération n’est effectuée.
- **Drop and create table** : la table est supprimée et créée à nouveau.
- **Create table** : la table n’existe pas et est créée.
- **Create table if does not exist** : la table est créée si elle n’existe pas.
- **Drop table if exist and create** : la table est supprimée si elle existe déjà et créée à nouveau.
- **Clear table** : le contenu de la table est supprimé. Vous pouvez annuler cette opération. |
### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center ([https://help.talend.com](https://help.talend.com)).

Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre *[Repository Content]*.

### Use schema columns for Copy

Cochez cette case pour utiliser l’option de colonne dans l'instruction COPY afin de restreindre la charge à une ou plusieurs colonne(s) spécifiée(s) dans la table. Pour plus d’informations, consultez *Vertica COPY SQL Statement* (en anglais).

### File Name

Chemin d’accès au fichier duquel charger les données. Le fichier doit se situer sur la machine où le Studio est installé ou celle où le Job utilisant ce composant est déployé.

Cette propriété est disponible uniquement lorsqu’il n’y a pas de flux d’entrée.

### Compression mode

 Sélectionnez le mode de compression pour le fichier duquel charger les données.

Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure.
<table>
<thead>
<tr>
<th><strong>Advanced settings</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Additional JDBC Parameters</strong></td>
<td>Spécifiez des paramètres JDBC supplémentaires pour la connexion à la base de données en cours de création. Cette propriété n’est pas disponible lorsque la case <strong>Use an existing connection</strong> est cochée.</td>
</tr>
</tbody>
</table>
| **Action on data** | Sélectionnez une action à effectuer sur les données de la table définie.  
  - **Bulk insert** : insère plusieurs lignes dans la table en une fois, au lieu d’insérer les lignes une par une. Si des doublons sont trouvés, le Job s’arrête.  
  - **Bulk update** : effectue plusieurs mises à jour simultanées sur différentes lignes. |
<p>| <strong>Stream name</strong> | Nom du flux d’un chargement, permettant d’identifier un chargement particulier. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure. |
| <strong>Write to ROS (Read Optimized Store)</strong> | Cochez cette case pour stocker les données dans une zone de chargement physique, pour en optimiser la lecture, lors de la compression et du pré-tri des données. |
| <strong>Exit Job on no rows loaded</strong> | Le Job s’arrête automatiquement si aucune ligne n’est chargée. |
| <strong>Missing columns as null</strong> | Cochez cette case pour insérer des valeurs NULL pour les colonnes manquantes lorsque les données sont insuffisantes pour correspondre aux colonnes spécifiées dans le schéma. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure. |
| <strong>Skip Header</strong> | Cochez cette case et, dans le champ qui s’affiche, spécifiez le nombre d’enregistrements à ignorer dans le fichier. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure. |
| <strong>Record terminator</strong> | Cochez cette case et, dans le champ qui s’affiche, spécifiez la chaîne de caractères littéraux utilisée pour indiquer la fin de chaque enregistrement dans le fichier. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure. |
| <strong>Enclosed by character</strong> | Cochez cette case pour configurer le caractère entourant les données. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure. |
| <strong>Escape char</strong> | Cochez cette case et, dans le champ qui s’affiche, spécifiez le caractère d’échappement pour le... |</p>
<table>
<thead>
<tr>
<th><strong>Fields terminated by</strong></th>
<th>Caractère, chaîne de caractères ou expression régulière pour séparer les champs.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Null String</strong></td>
<td>Chaîne de caractères affichée pour indiquer que la valeur est nulle.</td>
</tr>
<tr>
<td><strong>Reject not fitted values</strong></td>
<td>Cochez cette case pour rejeter les lignes de données de type char, varchar, binary et varbinary si elles ne vont pas dans la table cible. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure.</td>
</tr>
<tr>
<td><strong>Maximum number of rejected records</strong></td>
<td>Cochez cette case et, dans le champ qui s’affiche, spécifiez le nombre maximal d’enregistrements pouvant être rejetés avant échec du chargement. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure.</td>
</tr>
<tr>
<td><strong>Stop and rollback if any row is rejected</strong></td>
<td>Cochez cette case pour arrêter un chargement et effectuer un rollback sans charger de données si la moindre ligne est rejetée. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure.</td>
</tr>
<tr>
<td><strong>Don’t commit</strong></td>
<td>Cochez cette case pour effectuer une transaction de chargement de masse sans commiter automatiquement les résultats. Cette option est utile si vous souhaitez exécuter plusieurs chargements de masse en une seule transaction. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure.</td>
</tr>
<tr>
<td><strong>Rejected data file</strong></td>
<td>Spécifiez le fichier dans lequel écrire les lignes rejetées. Cette propriété est disponible uniquement lorsque l’action Bulk insert est sélectionnée dans la liste Action on data.</td>
</tr>
<tr>
<td><strong>Exception log file</strong></td>
<td>Spécifiez le fichier dans lequel écrire le log d’erreur. Ce log explique pourquoi chaque ligne rejetée est rejetée. Cette propriété est disponible uniquement lorsque l’action Bulk insert est sélectionnée dans la liste Action on data.</td>
</tr>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.</td>
</tr>
</tbody>
</table>
Variables globales

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCEPTED_ROW_NUMBER</td>
<td>Nombre de lignes à charger dans la base de données. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>REJECTED_ROW_NUMBER</td>
<td>Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

Utilisation

Règle d’utilisation

Le Studio Talend et la base de données Vertica permettent de créer des applications d’entrepôt de données et de datamart très rapides et abordables. Pour plus d’informations sur la manière de configurer le Studio Talend pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

Vous pouvez utiliser ce composant de l’une des manières suivantes pour écrire des données dans Vertica.

- il peut être utilisé en standalone dans un sous-job pour écrire des données dans Vertica à partir d’un fichier généré par un tVerticaOutputBulk.
- vous pouvez relier un composant tFileInputRaw à ce composant à l’aide d’un lien Row > Main pour écrire des données dans Vertica. Ainsi, le tFileInputRaw doit être en mode Stream the file et doit contenir une colonne de type Object dans son schéma.

Dynamic settings

Cliquez sur le bouton [+ ] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à
Scénarios associés

Pour un scénario associé au composant tVerticaBulkExec, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du composant tMysqlOutputBulk.
- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du composant tMysqlOutputBulkExec.
- **Scénario : Supprimer et insérer des données dans une base Oracle** à la page 2914 du composant tOracleBulkExec.
**tVerticaClose**

Ce composant ferme une connexion à la base de données Vertica.

**Propriétés du tVerticaClose Standard**

Ces propriétés sont utilisées pour configurer le tVerticaClose s’exécutant dans le framework de Jobs Standard.

Le composant tVerticaClose Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component List</td>
<td>Sélectionnez le composant dans lequel la connexion a été configurée.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

**Utilisation**

**Règle d’utilisation**

Ce composant est généralement utilisé avec d’autres composants Vertica, notamment avec le **tVerticaConnection** et le **tVerticaCommit**.

Le Studio Talend et la base de données Vertica permettent de créer des applications d’entrepôt de données et de datamart très rapides et abordables. Pour plus d’informations sur la manière de configurer le Studio Talend pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ **Code**, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez...
accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
**tVerticaCommit**

Ce composant ommite en une seule fois une transaction globale, à l’aide d’une connexion unique, au lieu de commiter chaque ligne ou chaque lot de lignes. Ce composant permet un gain de performance.

**Propriétés du tVerticaCommit Standard**

Ces propriétés sont utilisées pour configurer le tVerticaCommit s’exécutant dans le framework de Jobs Standard.

Le composant tVerticaCommit Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

![Remarque :](image)

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component List</td>
<td>Sélectionnez le composant dans lequel la connexion a été configurée.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Cochez cette case pour fermer la connexion à la base de données une fois que le composant a terminé sa tâche. Décochez cette case pour continuer à utiliser la connexion sélectionnée, une fois que le composant a terminé sa tâche. Si ce composant est lié à votre Job par un lien Row &gt; Main, vos données seront commitées ligne par ligne. Dans ce cas, ne cochez pas la case Close connection ou votre connexion sera fermée avant la fin du commit de la première ligne.</td>
</tr>
</tbody>
</table>

### Advanced settings

| tStatCatcher Statistics   | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Utilisation

| Règle d’utilisation       | Ce composant est généralement utilisé avec des composants Vertica et notamment avec les composants tVerticaConnection et tVerticaRollback. |
**Le Studio Talend** et la base de données Vertica permettent de créer des applications d’entrepôt de données et de datamart très rapides et abordables.

Pour plus d’informations sur la manière de configurer le Studio Talend pour se connecter à Vertica, consultez *Talend and HP Vertica Tips and Techniques* (en anglais).

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez *Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte* à la page 2641 et *Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement* à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

**Scénario associé**

Pour plus d’informations relatives au fonctionnement du composant tVerticaCommit, consultez *Scénario : Insérer des données dans des tables mère/fille* à la page 2620.
tVerticaConnection

Ce composant ouvre une connexion à la base de données spécifiée afin de pouvoir la réutiliser dans le(s) sous-job(s) suivant(s).

Propriétés du tVerticaConnection Standard

Ces propriétés sont utilisées pour configurer le tVerticaConnection s’exécutant dans le framework de Jobs Standard.

Le composant tVerticaConnection Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td>• Built-In : les informations de connexion seront stockés localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
<td></td>
</tr>
<tr>
<td>• Repository : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
<td></td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version de la base de données.</td>
</tr>
<tr>
<td>Host</td>
<td>Saisissez l’adresse IP ou le nom d’hôte de la base de données.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute de la base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Schéma de la base de données.</td>
</tr>
</tbody>
</table>
### Username et Password
Saisissez les données d’authentification de l’utilisateur de la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.

### Additional JDBC Parameters
Spécifiez des paramètres JDBC supplémentaires pour la connexion à la base de données en cours de création.

### Use or register a shared DB Connection
Cochez cette case afin de partager votre connexion à la base de données ou récupérer une connexion partagée par un Job père ou fils. Dans le champ Shared DB Connection Name qui s’affiche, saisissez un nom pour la connexion à la base de données partagée. Cela vous permet de partager une connexion à une base de données (à l’exception du paramètre de schéma de la base de données) à plusieurs composants de connexion, à différents niveaux de Jobs, enfants ou parents.

Cette option est incompatible avec les options Use dynamic job et Use an independent process to run subjob du composant tRunJob. Utiliser une connexion partagée avec un composant tRunJob ayant une de ces options activée fera échouer votre Job.

### Advanced settings

#### Auto Commit
Cochez cette case afin de commiter automatiquement toute modification dans la base de données lorsque la transaction est terminée.

Lorsque cette case est cochée, vous ne pouvez utiliser les composants de commit correspondant pour commiter les modifications dans la base de données. De la même manière, lorsque vous utilisez un composant de commit, cette case doit être décochée. Par défaut, la fonctionnalité d’auto-commit est désactivée et les modifications doivent être commises de manière explicite à l’aide du composant correspondant de commit.

Notez que la fonctionnalité d’auto-commit permet de commiter chaque instruction SQL comme transaction unique immédiatement après son exécution et que le composant de commit ne commite pas jusqu’à ce que toutes les instructions soient exécutées. Pour cette raison, si vous avez besoin de plus d’espace pour gérer vos transactions dans un Job, il est recommandé d’utiliser un composant Commit.

#### tStatCatcher Statistics
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.
## Utilisation

| Règle d'utilisation | Ce composant est généralement utilisé avec des composants Vertica, notamment les composants `tVerticaCommit` et `tVerticaRollback`. Le Studio Talend et la base de données Vertica permettent de créer des applications d'entrepôt de données et de datamart très rapides et abordables. Pour plus d'informations sur la manière de configurer le Studio Talend pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais). |

## Scénario associé

Pour un scénario associé au composant `tVerticaConnection`, consultez `tMysqlConnection` à la page 2618.
tVerticalInput

Ce composant récupère des données d’une table d’une base de données Vertica en se basant sur une requête SQL.

**Propriétés du tVerticalInput Standard**

Ces propriétés sont utilisées pour configurer le tVerticalInput s’exécutant dans le framework de Jobs Standard.

Le composant tVerticalInput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
</table>
| Property Type | Sélectionnez la manière de configurer les informations de connexion.  
• Built-In : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.  
• Repository : les informations de connexion stockées centralement dans le Repository > Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées. |
| ![icon] | Cliquez sur l’icône pour ouvrir l’assistant de connexion à une base de données et stocker les paramètres de connexion à la base de données configurés dans le composant.  
Pour plus d’informations concernant la configuration et le stockage des paramètres de connexion, consultez le Guide utilisateur du Studio Talend. |
| DB Version | Sélectionnez la version de la base de données. |
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List |
pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, si vous devez partager une connexion existante entre ces deux niveaux, par exemple pour partager la connexion créée par le Job père au Job fils, vous devez :

1. au niveau du Job père, enregistrer la connexion à la base de données à partager dans la vue **Basic settings** du composant de connexion créant cette connexion à la base de données.
2. au niveau du Job fils, utiliser un composant de connexion dédié afin de lire cette connexion enregistrée.

Pour un exemple concernant le partage de connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th>Host</th>
<th>Saisissez l’adresse IP ou le nom d’hôte de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute de la base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Schéma de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les données d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
</tbody>
</table>
| Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.  
- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le **Guide utilisateur du Studio Talend**.  
- **Repository** : Le schéma existe déjà et il est stocké dans le **Repository**. Ainsi, il peut être réutilisé. Voir également le **Guide utilisateur du Studio Talend**.  
Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.  
- **View schema** : sélectionnez cette option afin de voir le schéma.  
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales. |
• **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

| Table Name | Saisissez le nom de la table de laquelle récupérer les données. |
| Query Type et Query | Saisissez votre requête de base de données en faisant attention à ce que l’ordre des champs corresponde à celui défini dans le schéma.  
• **Built-In** : saisissez la requête manuellement dans le champ **Query** ou cliquez sur le bouton [...] à côté du champ **Query** afin de construire l’instruction graphiquement, à l’aide du SQLBuilder.  
• **Repository** : sélectionnez la requête correspondante stockée dans le référentiel en cliquant sur le bouton [...]. Dans la boîte de dialogue **Repository Content**, sélectionnez la requête à utiliser. Le champ **Query** est automatiquement renseigné. |

### Advanced settings

| Additional JDBC Parameters | Spécifiez des paramètres JDBC supplémentaires pour la connexion à la base de données en cours de création. Cette propriété n’est pas disponible lorsque la case **Use an existing connection** est cochée. |
| Trim all the String/Char columns | Cochez cette case pour supprimer les espaces en début et en fin de champ dans toutes les colonnes de type String/Char. |
| Trim column | Cochez cette case pour la colonne dont vous souhaitez supprimer les espaces en début et fin de champ. Cette propriété n’est pas disponible lorsque la case **Trim all the String/Char columns** est cochée. |
| tStatCatcher Statistics | Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant. |

### Global Variables

| NB_LINE | Nombre de lignes traitées. Cette variable est une variable After et retourne un entier. |
| QUERY | Instruction de requête en cours de traitement. Cette variable est une variable Flow et retourne une chaîne de caractères. |
**ERROR_MESSAGE**

Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.

**Utilisation**

| Règle d’utilisation | Ce composant couvre toutes les possibilités de requête SQL pour la base de données Vertica.

Le *Studio Talend* et la base de données Vertica permettent de créer des applications d’entrepôt de données et de datamart très rapides et abordables.

Pour plus d’informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez *Talend and HP Vertica Tips and Techniques* (en anglais). |
| --- | --- |

**Dynamic settings**

Cliquez sur le bouton [*] pour ajouter une ligne à la table. Dans le champ *Code*, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table *Dynamic settings* est disponible uniquement lorsque la case *Use an existing connection* est cochée dans la vue *Basic settings*. Lorsqu’un paramètre dynamique est configuré, la liste *Component List* de la vue *Basic settings* devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez *Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte* à la page 2641 et *Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement* à la page 520. Pour plus d’informations concernant les paramètres dynamiques et les variables de contexte, consultez le *Guide utilisateur du Studio Talend*.

**Scénarios associés**

Pour un scénario associé, consultez :

- *Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement* à la page 520.
Ce composant insère, met à jour, supprime, modifie ou copie des données d’un flux entrant dans une table d’une base de données Vertica.

Propriétés du tVerticaOutput Standard

Ces propriétés sont utilisées pour configurer le tVerticaOutput s’exécutant dans le framework de Jobs Standard.

Le composant tVerticaOutput Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• Built-In : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• Repository : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Cliquez sur l’icône pour ouvrir l’assistant de connexion à une base de données et stocker les paramètres de connexion à la base de données configurés dans le composant.</td>
</tr>
<tr>
<td></td>
<td>Pour plus d’informations concernant la configuration et le stockage des paramètres de connexion, consultez le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version de la base de données.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List</td>
</tr>
</tbody>
</table>
pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, si vous devez partager une connexion existante entre ces deux niveaux, par exemple pour partager la connexion créée par le Job père au Job fils, vous devez :

1. au niveau du Job père, enregistrer la connexion à la base de données à partager dans la vue **Basic settings** du composant de connexion créant cette connexion à la base de données.

2. au niveau du Job fils, utiliser un composant de connexion dédié afin de lire cette connexion enregistrée.

Pour un exemple concernant le partage de connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

<table>
<thead>
<tr>
<th><strong>Host</strong></th>
<th>Saisissez l’adresse IP ou le nom d’hôte de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>Saisissez le numéro du port d’écoute de la base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Schéma de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez les données d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Saisissez le nom de la table dans laquelle écrire les données.</td>
</tr>
</tbody>
</table>
| **Action on table** | Sélectionnez une opération à effectuer sur la table définie, vous pouvez effectuer l’une des opérations suivantes :
- **Default** : aucune opération n’est effectuée.
- **Drop and create table** : la table est supprimée et créée à nouveau.
- **Create table** : la table n’existe pas et est créée.
- **Create table if does not exist** : la table est créée si elle n’existe pas.
- **Drop table if exist and create** : la table est supprimée si elle existe déjà et créée à nouveau.
- **Clear table** : le contenu de la table est supprimé. Vous pouvez annuler cette opération. |
Use "drop cascade" | Cochez cette case pour supprimer tous les objets relatifs à la table qui sera supprimée.
Cette propriété est disponible uniquement lorsqu’une option relative à une suppression de table est sélectionnée dans la liste Action on table.

Action on data | Sélectionnez une action à effectuer sur les données de la table définie.
- Insert : ajoute de nouvelles entrées à la table. Le Job s’arrête lorsqu’il détecte des doublons.
- Update : met à jour les entrées existantes.
- Insert or update : insère un nouvel enregistrement. Si l’enregistrement avec la référence donnée existe déjà, une mise à jour est effectuée.
- Update or insert : met à jour l’enregistrement avec la référence donnée. Si l’enregistrement n’existe pas, un nouvel enregistrement est inséré.
- Delete : supprime les entrées correspondantes au flux d’entrée.
- Copy : lit des données d’un fichier texte et insère des tuples d’entrée dans le WOS (Write Optimized Store) ou directement dans le ROS (Read Optimized Store). Cette option convient au chargement de masse. Pour plus d’informations, consultez le Guide de référence Vertica SQL.

Il est nécessaire de spécifier au moins une colonne comme clé primaire sur laquelle baser les opérations Update et Delete. Pour ce faire, cliquez sur Edit Schema et cochez les cases à côté des colonnes que vous souhaitez définir comme clés primaires. Pour une utilisation avancée, cliquez sur l’onglet Advanced settings, vue dans laquelle vous pouvez définir simultanément des clés primaires pour les opérations Update et Delete. Pour ce faire, cochez la case Use field options et, dans la colonne Key in update column, cochez les cases à côté des colonnes à utiliser comme base pour l’opération Update. Répétez l’opération dans la colonne Key in delete column pour l’opération Delete.

Schema et Edit schema | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.
- Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces
valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur Edit schema pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Advanced settings

#### Additional JDBC Parameters

Spécifiez des paramètres JDBC supplémentaires pour la connexion à la base de données en cours de création.

Cette propriété n’est pas disponible lorsque la case **Use an existing connection** est cochée.

#### Abort on error

Cochez cette case pour arrêter l’opération Copy si une ligne est rejetée et effectuer un rollback de l’opération. Aucune donnée n’est chargée.

Cette propriété est disponible uniquement lorsque l’action **COPY** est sélectionnée dans la liste **Action on data**.

#### Maximum rejects

Saisissez un nombre pour configurer la commande REJECTMAX utilisée par Vertica, qui indique la limite maximale d’enregistrements logiques rejetés avant échec du chargement. Si cette option n’est pas renseignée ou que sa valeur est 0, un nombre illimité de rejets est autorisé.
| **No commit** | Cochez cette case pour empêcher la transaction en cours d'être committée automatiquement. Cette propriété est disponible uniquement lorsque l'action COPY est sélectionnée dans la liste Action on data. |
| **Exception file** | Saisissez le chemin d'accès au fichier ou parcourez votre système jusqu'au fichier dans lequel écrire les messages, indiquant le numéro de la ligne d'entrée et la raison pour le rejet de chaque enregistrement de données. Cette propriété est disponible uniquement lorsque l'action COPY est sélectionnée dans la liste Action on data. |
| **Exception file node** | Saisissez le nom du nœud pour le fichier des exceptions. S'il n'est pas spécifié, les opérations se font par défaut sur le nœud ayant initié la requête. Cette propriété est disponible uniquement lorsque l'action COPY est sélectionnée dans la liste Action on data. |
| **Rejected data file** | Saisissez le chemin d'accès au fichier ou parcourez votre système jusqu'au fichier dans lequel écrire les lignes rejetées. Ce fichier peut être modifié afin de résoudre les problèmes et rechargé. Cette propriété est disponible uniquement lorsque l'action COPY est sélectionnée dans la liste Action on data. |
| **Rejected data file node** | Saisissez le nom du nœud du fichier des données rejetées. S'il n'est pas spécifié, les opérations se font par défaut sur le nœud ayant initié la requête. Cette propriété est disponible uniquement lorsque l'action COPY est sélectionnée dans la liste Action on data. |
| **Commit every** | Spécifiez le nombre de lignes à traiter avant de commiter un lot de lignes dans la base de données. Cette option assure la qualité de la transaction (mais pas de rollback) et de meilleures performances lors de l'exécution. |
| **Use batch mode** | Cochez cette case pour activer le mode de traitement par lots pour le traitement des données et, dans le champ Batch size qui s'affiche, spécifiez le nombre d'enregistrements à traiter dans chaque lot. Cette propriété est disponible uniquement lorsque l'option Insert, Update, Delete ou Copy est sélectionnée dans la liste Action on data. |
| **Additional Columns** | Cette option vous permet d'appeler des fonctions SQL afin d'effectuer des actions sur des colonnes, actions |
qui ne sont ni des insertions, ni des mises à jour, ni des suppressions, ni des actions demandant un pré-traitement particulier. Cette option ne vous est pas proposée si vous créez (avec ou sans suppression) une table de base de données.

- **Name**: saisissez le nom de la colonne du schéma à modifier ou à insérer en tant que nouvelle colonne.
- **DataType**: saisissez le type de données pour la nouvelle colonne.
- **SQL expression**: saisissez l'instruction SQL à exécuter afin de modifier ou d’insérer les données de la colonne souhaitée.
- **Position**: sélectionnez Before, Replace ou After selon l’action à effectuer sur la colonne de référence.
- **Reference column**: sélectionnez une colonne de référence que le composant peut utiliser pour placer ou remplacer la nouvelle colonne ou la colonne modifiée.

### Use field options
Cochez la case des colonnes correspondantes afin de personnaliser une requête, particulièrement si plusieurs actions sont effectuées sur les données.

- **Update Key**: cochez la case de la colonne à partir de laquelle les données sont mises à jour.
- **Deletion Key**: cochez la case de la colonne à partir de laquelle les données sont supprimées.
- **Updatable**: cochez cette case si les données de la colonne correspondante peuvent être mises à jour.
- **Insertable**: cochez cette case si les données de la colonne correspondante peuvent être insérées.

### Enable debug mode
Cochez cette case pour afficher chaque étape lors du traitement des données dans une base de données.

### Support null in "SQL WHERE" statement
Cochez cette case pour valider la valeur Null dans l'instruction "SQL WHERE".

### Create projection when create table
Cochez cette case afin de créer une projection pour une table à créer.
Cette case est disponible uniquement lorsqu’une option relative à la création de table est sélectionnée dans la liste **Action on table**.

### tStatCatcher Statistics
Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th><strong>NB_LINE</strong></th>
<th>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NB_LINE_COPIED</td>
<td>Nombre de lignes copiées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_DELETED</td>
<td>Nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_INSERTED</td>
<td>Nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_REJECTED</td>
<td>Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_UPDATED</td>
<td>Nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

**Règle d’utilisation**

Ce composant est généralement utilisé comme composant de sortie. Il permet de faire des actions sur une table ou sur les données d’une table d’une base de données Vertica. Il permet aussi de créer un flux de rejet avec un lien Row > Rejects filtrant les données en erreur. Pour un exemple d’utilisation, consultez Scénario : Récupérer les données erronées à l’aide d’un lien Reject à la page 2675 du composant tMysqlOutput.

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.

Pour des exemples relatifs à l’utilisation des paramètres dynamiques, consultez Scénario : Lire des données dans des bases de données à l’aide de connexions dynamiques basées sur les variables de contexte à la page 2641 et Scénario : Lire des données à partir de différentes bases de données MySQL à l’aide de paramètres de connexion chargés dynamiquement à la page 520. Pour plus d’informations concernant les
Scénarios associés

Pour un scénario associé au composant **tVerticaOutput**, consultez :

- **Scénario : Insérer une colonne et modifier les données en utilisant le tMysqlOutput** à la page 2667 du composant **tMysqlOutput**.
**tVerticaOutputBulk**

Prépare un fichier à utiliser par le composant tVerticaBulkExec pour alimenter une base de données Vertica.

Les composants tVerticaOutputBulk et tVerticaBulkExec sont généralement utilisés ensemble comme deux parties d’un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l’opération d’INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tVerticaOutputBulkExec. L’avantage d’utiliser deux composants séparés réside dans le fait de pouvoir transformer les données avant leur chargement dans la base de données.

**Propriétés du tVerticaOutputBulk Standard**

Ces propriétés sont utilisées pour configurer le tVerticaOutputBulk s’exécutant dans le framework de Jobs Standard.

Le composant tVerticaOutputBulk Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez *Composants de bases de données dynamiques* à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Type</strong></td>
<td>Sélectionnez la manière dont le chemin d’accès au fichier et le schéma seront configurés.</td>
</tr>
<tr>
<td>- Built-In</td>
<td>le chemin d’accès au fichier et le schéma seront configurés localement pour ce composant.</td>
</tr>
<tr>
<td>- Repository</td>
<td>les détails du fichier stockés centralement dans le Repository &gt; Metadata seront réutilisés par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content qui s’ouvre, sélectionnez le fichier à réutiliser. Toutes les propriétés relatives seront automatiquement renseignées.</td>
</tr>
<tr>
<td><strong>File Name</strong></td>
<td>Chemin d’accès au fichier à géné rer. Ce fichier est généré sur la machine où est installé votre Studio ou sur celle où votre Job utilisant ce composant est déployé.</td>
</tr>
</tbody>
</table>
### Append

Cochez cette case pour ajouter de nouvelles lignes à la fin du fichier.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In**: Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center (https://help.talend.com).

Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.

- **View schema**: sélectionnez cette option afin de voir le schéma.
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### Advanced settings

<table>
<thead>
<tr>
<th>Row Separator</th>
<th>Saisissez le séparateur utilisé pour identifier la fin d'une ligne.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include Header</td>
<td>Cochez cette case pour inclure l’en-tête de la colonne dans le fichier.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Sélectionnez l’encodage dans la liste ou sélectionnez <strong>CUSTOM</strong> et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.</td>
</tr>
</tbody>
</table>
**tStatCatcher Statistics**

Cochez cette case pour collecter les métdonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE</td>
<td>Nombre de lignes traitées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

### Utilisation

**Règle d’utilisation**

Ce composant est généralement utilisé avec le composant `tVerticaBulkExec`. Ensemble, ils offrent un gain de performance important pour l’alimentation d’une base de données Vertica.

Le *Studio Talend* et la base de données Vertica permettent de créer des applications d’entrepôt de données et de datamart très rapides et abordables.

Pour plus d’informations sur la manière de configurer le *Studio Talend* pour se connecter à Vertica, consultez *Talend and HP Vertica Tips and Techniques* (en anglais).

### Scénarios associés

Pour un scénario associé au `tVerticaOutputBulk`, consultez :

- **Scénario : Insérer des données transformées dans une base MySQL** à la page 2685 du composant `tMysqlOutputBulk`.
- **Scénario : Insérer des données en masse dans une base MySQL** à la page 2692 du `tMysqlOutputBulkExec`. 
tVerticaOutputBulkExec

Ce composant reçoit des données du composant précédent, écrit les données dans un fichier local et les charge dans une base de données Vertica, à partir du fichier, à l’aide d’une instruction Vertica COPY SQL.

Pour plus d’informations concernant l'instruction Vertica COPY SQL, consultez COPY (en anglais).

Les composants tVerticaOutputBulk et tVerticaBulkExec sont généralement utilisés ensemble comme deux parties d’un processus en deux étapes. Dans la première étape, un fichier de sortie est généré. Dans la deuxième étape, ce fichier est utilisé lors de l’opération d'INSERT afin de peupler une base de données. Cette exécution en deux étapes est unifiée dans le composant tVerticaOutputBulkExec. L’avantage d’utiliser deux composants séparés réside dans le fait de pouvoir transformer les données avant leur chargement dans la base de données.

Propriétés du tVerticaOutputBulkExec Standard

Ces propriétés sont utilisées pour configurer le tVerticaOutputBulkExec s’exécutant dans le framework de Jobs Standard.

Le composant tVerticaOutputBulkExec Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

Remarque :

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td>• Built-In : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
<td></td>
</tr>
<tr>
<td>• Repository : les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
<td></td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version de la base de données.</td>
</tr>
</tbody>
</table>
| Use an existing connection | Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste **Component List** pour réutiliser les paramètres d’une connexion que vous avez déjà définie.

Notez que lorsqu’un Job contient un Job parent et un Job enfant, si vous devez partager une connexion existante entre ces deux niveaux, par exemple pour partager la connexion créée par le Job père au Job fils, vous devez :

1. au niveau du Job père, enregistrer la connexion à la base de données à partager dans la vue **Basic settings** du composant de connexion créant cette connexion à la base de données.

2. au niveau du Job fils, utiliser un composant de connexion dédié afin de lire cette connexion enregistrée.

Pour un exemple concernant le partage de connexion à travers différents niveaux de Jobs, consultez le **Guide utilisateur du Studio Talend**.

| Host | Saisissez l’adresse IP ou le nom d’hôte de la base de données. |
| Port | Saisissez le numéro du port d’écoute de la base de données. |
| DB Name | Saisissez le nom de la base de données. |
| Schema | Schéma de la base de données. |
| Username et Password | Saisissez les données d’authentification de l’utilisateur de la base de données.

Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |

| Action on data | Sélectionnez une action à effectuer sur les données de la table définie. |
| | • **Bulk insert** : insère plusieurs lignes dans la table en une fois, au lieu d’insérer les lignes une par une. Si des doublons sont trouvés, le Job s’arrête. |
| | • **Bulk update** : effectue plusieurs mises à jour simultanées sur différentes lignes. |
| Table | Saisissez le nom de la table dans laquelle écrire les données. |
| Action on table | Sélectionnez une opération à effectuer sur la table définie, vous pouvez effectuer l’une des opérations suivantes : |
| | • **Default** : aucune opération n’est effectuée. |
| | • **Drop and create table** : la table est supprimée et créée à nouveau. |
• **Create table**: la table n’existe pas et est créée.
• **Create table if does not exist**: la table est créée si elle n’existe pas.
• **Drop table if exist and create**: la table est supprimée si elle existe déjà et créée à nouveau.
• **Clear table**: le contenu de la table est supprimé. Vous pouvez annuler cette opération.

### Schema et Edit schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé `line` lors du nommage des champs.

- **Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.

Lorsque le schéma à réutiliser contient des valeurs par défaut de type Integer ou Function, assurez-vous que ces valeurs par défaut ne sont pas entourées de guillemets. Si elles le sont, supprimez manuellement les guillemets.

Vous pouvez obtenir plus de détails concernant la vérification des valeurs par défaut dans un schéma récupéré sur Talend Help Center ([https://help.talend.com](https://help.talend.com)).

Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

### File Name

Chemin d’accès au fichier à générer.
Ce fichier est généré sur la machine où est installé votre Studio ou sur celle où votre Job utilisant ce composant est déployé.

### Append

Cochez cette case pour ajouter de nouvelles lignes à la fin du fichier.
Use schema columns for Copy | Cochez cette case pour utiliser l’option de colonne dans l’instruction COPY afin de restreindre la charge à une ou plusieurs colonne(s) spécifiée(s) dans la table. Pour plus d’informations, consultez Vertica COPY SQL Statement (en anglais).

### Advanced settings

<p>| <strong>Additional JDBC Parameters</strong> | Spécifiez des paramètres JDBC supplémentaires pour la connexion à la base de données en cours de création. Cette propriété n’est pas disponible lorsque la case Use an existing connection est cochée. |
| <strong>Stream name</strong> | Nom du flux d’un chargement, permettant d’identifier un chargement particulier. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure. |
| <strong>Write to ROS (Read Optimized Store)</strong> | Cochez cette case pour stocker les données dans une zone de chargement physique, pour en optimiser la lecture, lors de la compression et du pré-tri des données. |
| <strong>Exit Job if no row was loaded</strong> | Le Job s’arrête automatiquement si aucune ligne n’est chargée. |
| <strong>Missing columns as null</strong> | Cochez cette case pour insérer des valeurs NULL pour les colonnes manquantes lorsque les données sont insuffisantes pour correspondre aux colonnes spécifiées dans le schéma. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure. |
| <strong>Skip Header</strong> | Cochez cette case et, dans le champ qui s’affiche, spécifiez le nombre d’enregistrements à ignorer dans le fichier. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure. |
| <strong>Record terminator</strong> | Cochez cette case et, dans le champ qui s’affiche, spécifiez la chaîne de caractères littéraux utilisée pour indiquer la fin de chaque enregistrement dans le fichier. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure. |
| <strong>Enclosed by character</strong> | Cochez cette case pour configurer le caractère entourant les données. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure. |
| <strong>Field Separator</strong> | Caractère, chaîne de caractères ou expression régulière pour séparer les champs. |
| <strong>Null String</strong> | Chaîne de caractères affichée pour indiquer que la valeur est nulle. |</p>
<table>
<thead>
<tr>
<th>Include Header</th>
<th>Cochez cette case pour inclure l’en-tête de la colonne dans le fichier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoding</td>
<td>Sélectionnez l’encodage dans la liste ou sélectionnez CUSTOM et définissez-le manuellement. Ce champ est obligatoire pour la gestion de données de bases de données.</td>
</tr>
<tr>
<td>Reject not fitted values</td>
<td>Cochez cette case pour rejeter les lignes de données de type char, varchar, binary et varbinary si elles ne vont pas dans la table cible. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure.</td>
</tr>
<tr>
<td>Maximum number of rejected records</td>
<td>Cochez cette case et, dans le champ qui s’affiche, spécifiez le nombre maximal d’enregistrements pouvant être rejetés avant échec du chargement. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure.</td>
</tr>
<tr>
<td>Stop and rollback if any row is rejected</td>
<td>Cochez cette case pour arrêter un chargement et effectuer un rollback sans charger de données si la moindre ligne est rejetée. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure.</td>
</tr>
<tr>
<td>Don’t commit</td>
<td>Cochez cette case pour effectuer une transaction de chargement de masse sans commiter automatiquement les résultats. Cette option est utile si vous souhaitez exécuter plusieurs chargements de masse en une seule transaction. Cette propriété est disponible uniquement lorsque vous utilisez Vertica 6.0 ou une version supérieure.</td>
</tr>
<tr>
<td>Rejected data file</td>
<td>Spécifiez le fichier dans lequel écrire les lignes rejetées. Cette propriété est disponible uniquement lorsque l’action Bulk insert est sélectionnée dans la liste Action on data.</td>
</tr>
<tr>
<td>Exception log file</td>
<td>Spécifiez le fichier dans lequel écrire le log d’erreur. Ce log explique pourquoi chaque ligne rejetée est rejetée. Cette propriété est disponible uniquement lorsque l’action Bulk insert est sélectionnée dans la liste Action on data.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<p>| ACCEPTED_ROW_NUMBER                                                        | Nombre de lignes à charger dans la base de données. Cette variable est une variable After et retourne un entier. |</p>
<table>
<thead>
<tr>
<th>REJECTED_ROW_NUMBER</th>
<th>Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

**Utilisation**

**Règle d’utilisation**

Ce composant est principalement utilisé lorsqu’aucune transformation particulière n’est requise sur les données à charger dans la base de données.

Le Studio Talend et la base de données Vertica permettent de créer des applications d’entrepôt de données et de datamart très rapides et abordables. Pour plus d’informations sur la manière de configurer le Studio Talend pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

**Scénarios associés**

Pour un scénario associé au **tVerticaOutputBulkExec**, consultez :

- Scénario : Insérer des données transformées dans une base MySQL à la page 2685 du composant **tMysqlOutputBulk**.

- Scénario : Insérer des données en masse dans une base MySQL à la page 2692 du **tMysqlOutputBulkExec**.
**tVerticaRollback**

Ce composant annule la transaction commitée dans la base de données Vertica connectée.

**Propriétés du tVerticaRollback Standard**

Ces propriétés sont utilisées pour configurer le tVerticaRollback s’exécutant dans le framework de Jobs Standard.

Le composant tVerticaRollback Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

⚠️ **Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Sélectionnez un type de base de données dans la liste et cliquez sur <strong>Apply</strong>.</td>
</tr>
<tr>
<td>Component List</td>
<td>Sélectionnez le composant dans lequel la connexion a été configurée.</td>
</tr>
<tr>
<td>Close Connection</td>
<td>Cochez cette case pour fermer la connexion à la base de données une fois que le composant a terminé sa tâche. Décochez cette case pour continuer à utiliser la connexion sélectionnée, une fois que le composant a terminé sa tâche.</td>
</tr>
</tbody>
</table>

### Advanced settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Utilisation

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Règle d’utilisation</td>
<td>Ce composant est généralement utilisé avec d’autres composants Vertica, notamment les composants <strong>tVerticaConnection</strong> et <strong>tVerticaCommit</strong>. Le <strong>Studio Talend</strong> et la base de données Vertica permettent de créer des applications d’entrepôt de données et de datamart très rapides et abordables. Pour plus d’informations sur la manière de configurer le <strong>Studio Talend</strong> pour se connecter à Vertica, consultez <strong>Talend and HP Vertica Tips and Techniques</strong> (en anglais).</td>
</tr>
</tbody>
</table>

4240
| Dynamic settings | Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d'un Studio Talend.

Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


---

**Scénario associé**

Pour un scénario associé au composant **tVerticaRollback**, consultez Scénario : Annuler l’insertion de données dans des tables mère/fille à la page 2623.
**tVerticaRow**

Ce composant exécute une instruction SQL Vertica sur une table d'une base de données.

### Propriétés du tVerticaRow Standard

Ces propriétés sont utilisées pour configurer le tVerticaRow s'exécutant dans le framework de Jobs Standard.

Le composant tVerticaRow Standard appartient à la famille Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d'un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d'informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

### Basic settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Type</strong></td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td>• <strong>Built-In</strong></td>
<td>les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td>• <strong>Repository</strong></td>
<td>les informations de connexion stockées centralement dans le Repository &gt; Metadata seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue Repository Content, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td><strong>DB Version</strong></td>
<td>Sélectionnez la version de la base de données.</td>
</tr>
<tr>
<td><strong>Use an existing connection</strong></td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste Component List pour réutiliser les paramètres d'une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td></td>
<td>Notez que lorsqu’un Job contient un Job parent et un Job enfant, si vous devez partager une connexion existante entre ces deux niveaux, par exemple pour partager la connexion créée par le Job père au Job fils, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. au niveau du Job père, enregistrer la connexion à la base de données à partager dans la vue Basic</td>
</tr>
</tbody>
</table>
settings du composant de connexion créant cette connexion à la base de données.

2. au niveau du Job fils, utiliser un composant de connexion dédié afin de lire cette connexion enregistrée.

Pour un exemple concernant le partage de connexion à travers différents niveaux de Jobs, consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Host</th>
<th>Saisissez l’adresse IP ou le nom d’hôte de la base de données.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Saisissez le numéro du port d’écoute de la base de données.</td>
</tr>
<tr>
<td>Database</td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td>Schema</td>
<td>Schéma de la base de données.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les données d’authentification de l’utilisateur de la base de données.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Schema et Edit schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>• Built-In : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>• Repository : Le schéma existe déjà et il est stocké dans le Repository. Ainsi, il peut être réutilisé. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
<tr>
<td></td>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
</tr>
<tr>
<td></td>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de proposer ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].</td>
</tr>
<tr>
<td><strong>Table Name</strong></td>
<td>Saisissez le nom de la table à traiter.</td>
</tr>
</tbody>
</table>
| **Query Type et Query** | Saisissez votre requête de base de données en faisant attention à ce que l'ordre des champs corresponde à celui défini dans le schéma.  
  - **Built-In** : saisissez la requête manuellement dans le champ **Query** ou cliquez sur le bouton [...] à côté du champ **Query** afin de construire l'instruction graphiquement, à l'aide du SQLBuilder.  
  - **Repository** : sélectionnez la requête correspondante stockée dans le référentiel en cliquant sur le bouton [...]. Dans la boîte de dialogue **Repository Content**, sélectionnez la requête à utiliser. Le champ **Query** est automatiquement renseigné. |
| **Die on error** | Cochez cette case pour arrêter l'exécution du Job lorsqu'une erreur survient.  
Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.  
Lorsque des erreurs sont ignorées, vous pouvez collecter les lignes en erreur à l'aide d'un lien **Row > Reject**. |

**Advanced settings**

| **Additional JDBC Parameters** | Spécifiez des paramètres JDBC supplémentaires pour la connexion à la base de données en cours de création. Cette propriété n'est pas disponible lorsque la case **Use an existing connection** est cochée. |
| **Propagate QUERY's recordset** | Cochez cette case afin de propager le résultat de la requête au flux de sortie. Dans la liste **use column** affichée, vous devez sélectionner une colonne dans laquelle insérer le résultat de la requête.  
Cette option permet au composant d'avoir un schéma différent de celui du composant précédent. De plus, la colonne contenant le résultat de la requête doit être de type Object. Ce composant est généralement suivi par un composant **tParseRecordSet**. |
| **Use PreparedStatement** | Cochez cette case si vous souhaitez interroger la base de données à l'aide d'une instruction préparée. Dans la table **Set PreparedStatement Parameters** qui s'affiche, spécifiez la valeur pour chaque paramètre représenté par un point d’interrogation ?, dans l'instruction SQL définie dans le champ **Query**.  
  - **Parameter Index** : position du paramètre dans l'instruction SQL.  
  - **Parameter Type** : type de données du paramètre.  
  - **Parameter Value** : valeur du paramètre.  
Pour un scénario utilisant cette propriété, consultez **Scénario 2 : Utiliser l’instance PreparedStatement pour faire une requête sur des données** à la page 2702. |
**Commit every**

Spécifiez le nombre de lignes à traiter avant de commiter un lot de lignes dans la base de données. Cette option assure la qualité de la transaction (mais pas de rollback) et de meilleures performances lors de l’exécution.

**tStatCatcher Statistics**

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

---

### Global Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB_LINE_DELETED</strong></td>
<td>Nombre de lignes supprimées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_INSERTED</strong></td>
<td>Nombre de lignes insérées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>NB_LINE_UPDATED</strong></td>
<td>Nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td><strong>QUERY</strong></td>
<td>Instruction de requête en cours de traitement. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>

---

### Utilisation

**Règle d’utilisation**

Ce composant offre la flexibilité des requêtes sur les bases de données et couvre toutes les possibilités de requêtes SQL.

Le Studio Talend et la base de données Vertica permettent de créer des applications d’entrepôt de données et de datamart très rapides et abordables. Pour plus d’informations sur la manière de configurer le Studio Talend pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

**Dynamic settings**

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée.
dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


Scénario associé

Pour un scénario associé au composant tVerticaRow, consultez :

- Scénario : Combiner deux flux pour une sortie sélective à la page 2706.
- Procédure du tDBSQLRow.
- Scénario : Supprimer et re-générer un index de table MySQL à la page 2700 du tMysqlRow.
**tVerticaSCD**

Ce composant traque et reflète les modifications d’une table Vertica SCD dédiée.

**Propriétés du tVerticaSCD Standard**

Ces propriétés sont utilisées pour configurer le tVerticaSCD s’exécutant dans le framework de Jobs Standard.

Le composant tVerticaSCD Standard appartient aux familles Business Intelligence et Databases.

Le composant de ce framework est toujours disponible.

**Remarque :**

Ce composant est une version spécifique d’un connecteur à une base de données dynamique. Les propriétés associées aux paramètres de la base de données dépendant du type de base de données sélectionné.

Pour plus d’informations concernant les connecteurs aux bases de données, consultez Composants de bases de données dynamiques à la page 634.

**Basic settings**

<table>
<thead>
<tr>
<th>Database</th>
<th>Sélectionnez un type de base de données dans la liste et cliquez sur Apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Type</td>
<td>Sélectionnez la manière de configurer les informations de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Built-In</strong> : les informations de connexion seront stockées localement pour ce composant. Vous devez spécifier manuellement les valeurs de toutes les propriétés de connexion.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Repository</strong> : les informations de connexion stockées centralement dans le <strong>Repository &gt; Metadata</strong> seront réutilisées par ce composant. Vous devez cliquer sur le bouton [...] et, dans la boîte de dialogue <strong>Repository Content</strong>, sélectionnez les détails de connexion à réutiliser. Toutes les propriétés de connexion seront automatiquement renseignées.</td>
</tr>
<tr>
<td>DB Version</td>
<td>Sélectionnez la version de la base de données.</td>
</tr>
<tr>
<td>Use an existing connection</td>
<td>Cochez cette case et sélectionnez le composant de connexion adéquat à partir de la liste <strong>Component List</strong> pour réutiliser les paramètres d’une connexion que vous avez déjà définie.</td>
</tr>
<tr>
<td></td>
<td>Notez que lorsqu’un Job contient un Job parent et un Job enfant, si vous devez partager une connexion existante entre ces deux niveaux, par exemple pour partager la connexion créée par le Job père au Job fils, vous devez :</td>
</tr>
<tr>
<td></td>
<td>1. au niveau du Job père, enregistrer la connexion à la base de données à partager dans la vue <strong>Basic</strong></td>
</tr>
</tbody>
</table>
**settings** du composant de connexion créant cette connexion à la base de données.

2. au niveau du Job fils, utiliser un composant de connexion dédié afin de lire cette connexion enregistrée.

Pour un exemple concernant le partage de connexion à travers différents niveaux de Jobs, consultez le *Guide utilisateur du Studio Talend*.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Saisissez l’adresse IP ou le nom d’hôte de la base de données.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Saisissez le numéro du port d’écoute de la base de données.</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>Saisissez le nom de la base de données.</td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td>Schéma de la base de données.</td>
</tr>
<tr>
<td><strong>Username et Password</strong></td>
<td>Saisissez les données d’authentification de l’utilisateur de la base de données. Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ <strong>Password</strong>, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur <strong>OK</strong> afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Saisissez le nom de la table dans laquelle écrire les données.</td>
</tr>
</tbody>
</table>
| **Action on table** | Sélectionnez une opération à effectuer sur la table définie, vous pouvez effectuer l’une des opérations suivantes :  
  - **None** : aucune opération n’est effectuée.  
  - **Create table** : la table n’existe pas et est créée.  
  - **Create table if does not exist** : la table est créée si elle n’existe pas. |
| **Schema et Edit schema** | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé *line* lors du nommage des champs.  
  - **Built-In** : le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le *Guide utilisateur du Studio Talend*.  

Cliquez sur **Edit schema** pour modifier le schéma. Notez que si vous effectuez des modifications, le schéma passe automatiquement en type built-in.  
  - **View schema** : sélectionnez cette option afin de voir le schéma. |
- **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.

- **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre *Repository Content*.

### SCD Editor
Cliquez sur le bouton [...] et, dans la boîte de dialogue *SCD component editor* qui s’ouvre, configurez le flux de données pour les sorties de dimension à évolution lente.

Pour plus d’informations, consultez *Méthodologie de gestion du SCD* à la page 2716.

### Use memory saving mode
Cochez cette case pour maximiser les performances du système.

### Die on error
Cochez cette case pour arrêter l’exécution du Job lorsqu’une erreur survient.

Décochez la case pour ignorer les lignes en erreur et terminer le processus avec les lignes sans erreur.

Lorsque des erreurs sont ignorées, vous pouvez collecter les lignes en erreur à l’aide d’un lien *Row > Reject*.

### Advanced settings

<table>
<thead>
<tr>
<th>Additional JDBC Parameters</th>
<th>Spécifiez des paramètres JDBC supplémentaires pour la connexion à la base de données en cours de création. Cette propriété n’est pas disponible lorsque la case <em>Use an existing connection</em> est cochée.</th>
</tr>
</thead>
<tbody>
<tr>
<td>End date time details</td>
<td>Spécifiez la valeur temporelle de la date de fin SCD au format <em>HH:mm:ss</em>. La valeur par défaut est <em>12:00:00</em>. Cette propriété est disponible uniquement lorsque la méthode SCD de Type 2 est utilisée et lorsque l’option <em>Fixed year value</em> est sélectionnée pour créer la date de fin SCD de Type 2 dans l’éditeur SCD.</td>
</tr>
<tr>
<td>Debug mode</td>
<td>Cochez cette case pour afficher chaque étape lors du traitement des données dans une base de données.</td>
</tr>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collection des métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

### Global Variables

| NB_LINE_INSERTED | Nombre de lignes insérées. Cette variable est une variable After et retourne un entier. |
## Utilisation

### Règle d’utilisation

Ce composant est généralement utilisé comme composant de fin dans un Job ou un sous-job et nécessite un lien d’entrée.

Le Studio Talend et la base de données Vertica permettent de créer des applications d’entrepôt de données et de datamart très rapides et abordables. Pour plus d’informations sur la manière de configurer le Studio Talend pour se connecter à Vertica, consultez Talend and HP Vertica Tips and Techniques (en anglais).

### Dynamic settings

Cliquez sur le bouton [+] pour ajouter une ligne à la table. Dans le champ Code, saisissez une variable de contexte afin de sélectionner dynamiquement votre connexion à la base de données parmi celles prévues dans votre Job. Cette fonctionnalité est utile si vous devez accéder à plusieurs tables de bases de données ayant la même structure mais se trouvant dans différentes bases de données, en particulier lorsque vous travaillez dans un environnement dans lequel vous ne pouvez pas changer les paramètres de votre Job, par exemple lorsque votre Job doit être déployé et exécuté indépendamment d’un Studio Talend.

La table Dynamic settings est disponible uniquement lorsque la case Use an existing connection est cochée dans la vue Basic settings. Lorsqu’un paramètre dynamique est configuré, la liste Component List de la vue Basic settings devient inutilisable.


### Limitation

Ce composant ne supporte pas l’utilisation du Type 0 de SCD avec d’autres Types de SCD.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB_LINE_UPDATED</td>
<td>Nombre de lignes mises à jour. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>NB_LINE_REJECTED</td>
<td>Nombre de lignes rejetées. Cette variable est une variable After et retourne un entier.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>Message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>
Scénario associé

Pour un scénario associé, consultez tMysqlSCD à la page 2712.
tVtigerCRMInput

Ce composant extrait les données d’un module d’une base de données VtigerCRM.

Propriétés du tVtigerCRMInput Standard

Ces propriétés sont utilisées pour configurer le tVtigerCRMInput s’exécutant dans le framework de Jobs Standard.

Le composant tVtigerCRMInput Standard appartient à la famille Business.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vtiger Version</td>
<td>Sélectionnez la version du Vtiger Web Services que vous souhaitez utiliser.</td>
</tr>
<tr>
<td>Server Address</td>
<td>Saisissez l’adresse IP du serveur VtigerCRM.</td>
</tr>
<tr>
<td>Port</td>
<td>Saisissez le numéro de port d’accès au serveur.</td>
</tr>
<tr>
<td>Vtiger Path</td>
<td>Saisissez le chemin d’accès au serveur.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d’authentification de l’utilisateur au service Web.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Version</td>
<td>Renseignez la version de VtigerCRM que vous utilisez.</td>
</tr>
<tr>
<td>Module</td>
<td>Sélectionnez le module adéquat dans la liste.</td>
</tr>
<tr>
<td>Method</td>
<td>Sélectionnez la méthode adéquate dans la liste. Cette méthode permet de déterminer l’action à appliquer au module VtigerCRM sélectionné.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.</td>
</tr>
<tr>
<td></td>
<td>Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
</tbody>
</table>
• **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode *Built-In* et effectuer des modifications locales.

• **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur *No* et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Pour ce composant, le schéma dépend du Module sélectionné.

Lorsque la version à utiliser est Vtiger 5.1 :

<table>
<thead>
<tr>
<th>Endpoint</th>
<th>Saisissez l’URL du serveur Web invoqué.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>Saisissez le nom d’utilisateur afin de vous connecter au VtigerCRM.</td>
</tr>
<tr>
<td>Access key</td>
<td>Saisissez la clé d’accès pour le nom d’utilisateur.</td>
</tr>
<tr>
<td>Query condition</td>
<td>Saisissez la requête permettant de sélectionner les données à extraire.</td>
</tr>
<tr>
<td>Manual input of SQL query</td>
<td>Cochez cette case pour saisir manuellement la requête que vous souhaitez effectuer dans le champ <em>Query</em>.</td>
</tr>
</tbody>
</table>

**Advanced settings**

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier. |
|------------------| **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option. |
|                  | Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant. |
|                  | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+Espace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

| Règle d’utilisation | Ce composant est généralement utilisé en tant que composant de début. Un composant de sortie est nécessaire. |

**Scénario associé**

Aucun scénario n’est disponible pour la version Standard de ce composant.
tVtigerCRMOoutput

Ce composant écrit des données dans un module d’une base de données VtigerCRM.

**Propriétés du tVtigerCRMOoutput Standard**

Ces propriétés sont utilisées pour configurer le tVtigerCRMOoutput s'exécutant dans le framework de Jobs Standard.

Le composant tVtigerCRMOoutput Standard appartient à la famille Business.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Vtiger Version</th>
<th>Sélectionnez la version du Vtiger Web Services que vous souhaitez utiliser.</th>
</tr>
</thead>
</table>

Lorsque la version à utiliser est Vtiger 5.0 :

<table>
<thead>
<tr>
<th>Server Address</th>
<th>Saisissez l’adresse IP du serveur VtigerCRM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Saisissez le numéro de port d’accès au serveur.</td>
</tr>
<tr>
<td>Vtiger Path</td>
<td>Saisissez le chemin d’accès au serveur.</td>
</tr>
<tr>
<td>Username et Password</td>
<td>Saisissez les informations d’authentification de l’utilisateur au service Web.</td>
</tr>
<tr>
<td></td>
<td>Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ Password, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur OK afin de sauvegarder les paramètres.</td>
</tr>
<tr>
<td>Version</td>
<td>Renseignez la version de VtigerCRM que vous utilisez.</td>
</tr>
<tr>
<td>Module</td>
<td>Sélectionnez le module adéquat dans la liste.</td>
</tr>
<tr>
<td>Method</td>
<td>Sélectionnez la méthode adéquate dans la liste. Cette méthode permet de déterminer l’action à appliquer au module VtigerCRM sélectionné.</td>
</tr>
<tr>
<td>Schema et Edit Schema</td>
<td>Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :</td>
</tr>
<tr>
<td></td>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
</tr>
</tbody>
</table>
Lorsque la version à utiliser est Vtiger 5.1 :

<table>
<thead>
<tr>
<th>Endpoint</th>
<th>Saisissez l’URL du serveur Web invoqué.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>Saisissez le nom d’utilisateur afin de vous connecter au VtigerCRM.</td>
</tr>
<tr>
<td>Access key</td>
<td>Saisissez la clé d’accès pour le nom d’utilisateur.</td>
</tr>
</tbody>
</table>
| Action         | Vous pouvez effectuer l’une des opérations suivantes sur les données du module VtigerCRM :
              | Insert : insérer les données. |
              | Update : mettre les données à jour. |
| Module         | Sélectionnez le module adéquat dans la liste. |

Schema et Edit Schema

Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content].

Pour ce composant, le schéma dépend du Module sélectionné.
### Die on error

Cette case est cochée par défaut et stoppe le Job en cas d'erreur. Décochez cette case pour terminer le traitement avec les lignes sans erreurs, et ignorer les lignes en erreur.

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | **NB_LINE** : nombre de lignes lues par un composant d'entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.
|                  | **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.
|                  | Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.
|                  | Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches *Ctrl+ESpace* pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
|                  | Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

### Utilisation

| Règle d’utilisation | Ce composant est utilisé comme composant de sortie. Il nécessite un composant d’entrée. |

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tWaitForFile

Ce composant fait une boucle sur un répertoire donné et déclenche le composant suivant lorsque la condition définie est rencontrée.

Le tWaitForFile est utilisé pour mettre en attente le composant lié. Il déclenche ce composant lorsque l’opération sur le fichier défini est effectuée dans le répertoire spécifié.

Propriétés du tWaitForFile Standard

Ces propriétés sont utilisées pour configurer le tWaitForFile s’exécutant dans le framework de Jobs Standard.

Le composant tWaitForFile Standard appartient à la famille Orchestration.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (in seconds) between iterations</td>
<td>Paramétrez l’intervalle de temps entre deux itérations (en secondes).</td>
</tr>
<tr>
<td>Max. number of iterations (infinite loop if empty)</td>
<td>Configurez le nombre maximal d’itérations à effectuer sur le répertoire spécifié.</td>
</tr>
<tr>
<td>Directory to scan</td>
<td>Spécifiez le répertoire sur lequel effectuer des itérations.</td>
</tr>
<tr>
<td>File mask</td>
<td>Saisissez le masque du fichier recherché afin de filtrer les fichiers à moniterer. Pour moniter un fichier, saisissez son nom.</td>
</tr>
<tr>
<td>Include subdirectories</td>
<td>Cochez cette case pour inclure les sous-répertoires dans l’itération.</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>Cochez cette case pour tenir compte de la casse.</td>
</tr>
<tr>
<td>Include present file</td>
<td>Cochez cette case pour inclure le fichier courant.</td>
</tr>
<tr>
<td>Trigger action when</td>
<td>Sélectionnez l’une des conditions de déclenchement :</td>
</tr>
<tr>
<td></td>
<td>- <strong>a file is created</strong> : déclenche le composant suivant lorsqu’un fichier est créé.</td>
</tr>
<tr>
<td></td>
<td>- <strong>a file is deleted</strong> : déclenche le composant suivant lorsqu’un fichier est supprimé.</td>
</tr>
<tr>
<td></td>
<td>- <strong>a file is updated</strong> : déclenche le composant suivant lorsqu’un fichier est modifié.</td>
</tr>
<tr>
<td></td>
<td>- <strong>a file is created or updated or deleted</strong> : déclenche le composant suivant lorsqu’un fichier est créé, supprimé ou modifié.</td>
</tr>
<tr>
<td>Then</td>
<td>Sélectionnez l’action à effectuer avec la boucle itérative après déclenchement du composant suivant :</td>
</tr>
<tr>
<td></td>
<td>- <strong>continue loop</strong> : continuer l’itération jusqu’à ce que le nombre maximum d’itérations soit atteint.</td>
</tr>
</tbody>
</table>
**Schema et Edit Schema**

<table>
<thead>
<tr>
<th>Schema et Edit Schema</th>
<th>Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma de ce composant est local (built-in) et ne peut être modifié.</th>
</tr>
</thead>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Wait for file to be released Every (in ms)</th>
<th>Cochez cette case afin que le composant suivant ne se déclenche qu’après la fin de l’opération.</th>
</tr>
</thead>
</table>

**Global Variables**

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
**CURRENT ITERATION** : numéro de séquence de l’itération courante. Cette variable est une variable Flow et retourne un entier.  
**PRESENT_FILE** : nom et chemin d’accès au fichier. Cette variable est une variable Flow et retourne une chaîne de caractères.  
**DELETED_FILE** : nom et chemin d’accès au fichier supprimé. Cette variable est une variable Flow et retourne une chaîne de caractères.  
**CREATED_FILE** : nom et chemin d’accès au fichier créé. Cette variable est une variable Flow et retourne une chaîne de caractères.  
**UPDATED_FILE** : nom et chemin d’accès au fichier mis à jour. Cette variable est une variable Flow et retourne une chaîne de caractères.  
**FILENAME** : nom du fichier traité. Cette variable est une variable Flow et retourne une chaîne de caractères.  
**NOT_UPDATED_FILE** : nom et chemin d’accès au fichier n’ayant pas été mis à jour. Cette variable est une variable Flow et retourne une chaîne de caractères. |
|------------------|---------------------------------------------------------------------------------------------------------------|

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. À partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |
Utilisation

| Règle d'utilisation | Ce composant est un composant de déclenchement du composant suivant, en fonction de la condition établie. Ainsi, ce composant doit être connecté à un autre composant. |

Connections

Scénario 1 : Attendre la création d’un fichier et arrêter l’itération après le déclenchement d’un message

Ce scénario décrit un Job scannant un répertoire et déclenchant un message affichant dans la console les informations de création d’un fichier lorsqu’un fichier texte est créé dans ce répertoire. Le scan prend fin et un autre message affiche le nombre d’itérations terminées.

Construire le Job

Procédure

2. Reliez le tWaitForFile au premier tJava à l’aide d’un lien Row > Iterate.
3. Reliez le tWaitForFile au second tJava à l’aide d’un lien Trigger > On Subjob Ok.

Configurer les composants

Procédure

1. Double-cliquez sur le tWaitForFile pour ouvrir sa vue Basic settings.
2. Dans le champ **Directory to scan**, saisissez le nom du répertoire à scanner.

3. Dans le champ **File mask**, saisissez “*.txt” pour scanner uniquement les fichiers texte.

4. Dans la liste déroulante **Trigger action when**, sélectionnez **a file is created** afin de déclencher le premier **Java** lorsqu’un fichier texte est créé.

5. Dans la liste déroulante **Then**, sélectionnez **exit loop** afin d’arrêter la boucle itérative dès que le premier **Java** a été déclenché.

6. Double-cliquez sur le premier **Java** pour ouvrir sa vue **Basic settings**.

7. Dans le champ **Code**, saisissez le code suivant :
   ```java
   System.out.println("A file was created at " + TalendDate.getCurrentDate());
   System.out.println("Name of the created file: " + ((String)globalMap.get("tWaitForFile_1_CREATED_FILE"));
   ```

8. Double-cliquez sur le second **Java** pour ouvrir sa vue **Basic settings**.

9. Dans le champ **code**, saisissez le code suivant :
   ```java
   System.out.println("\nIteration loop ended at " + TalendDate.getCurrentDate());
   System.out.println("Number of iterations finished: " + ((Integer)globalMap.get("tWaitForFile_1_CURRENT_ITERATION"));
   ```

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.
3. Créez un fichier texte **exit.txt** dans le répertoire spécifié.
Lorsqu’un fichier texte est créé, un message concernant la création du fichier apparaît dans la console de la vue Run. L’itération se termine immédiatement et est suivie d’un autre message affichant l’heure de fin de la boucle itérative et le nombre d’itérations effectuées.

Scénario 2 : Attendre la création d’un fichier et poursuivre l’itération après le déclenchement d’un message

Basé sur le scénario précédent, ce scénario décrit un Job déclenchant un message dans la console ainsi que la création d’un fichier texte dans le répertoire scanné, sans arrêter le processus de scan immédiatement. Le Job continue à scanner le répertoire jusqu’à ce que le nombre d’itérations atteigne la valeur maximale définie.

Procédure

1. Double-cliquez sur le tWaitForFile pour ouvrir sa vue Basic settings.

2. Dans le champ Time (in seconds) between iterations, saisissez l’intervalle de temps en secondes entre deux itérations. Dans cet exemple, saisissez 5.

3. Dans le champ Max. number of iterations (infinite loop if empty), saisissez le nombre maximal d’itérations. Dans cet exemple, saisissez 10.

4. Dans la liste Then, sélectionnez continue loop afin de continuer la boucle itérative, même après déclenchement du message.

5. Appuyez sur les touches Ctrl+S afin de sauvegarder le Job.

6. Appuyez sur F6 pour exécuter le Job.
7. Créez un fichier texte `continue.txt` dans le répertoire spécifié.

Lorsqu’un fichier texte est créé, un message concernant la création du fichier est affiché dans la console de la vue **Run**. La boucle itérative est arrêtée jusqu’à ce que le nombre d’itérations atteigne dix. Un nouveau message s’affiche et indique l’heure de fin de la boucle itérative et le nombre d’itérations effectuées.
**tWaitForSocket**

Ce composant déclenche un Job en fonction d’une condition définie.

Le tWaitForSocket fait une boucle sur un port défini à la recherche de données puis déclenche un sous-job lorsque la condition est rencontrée.

**Propriétés du tWaitForSocket Standard**

Ces propriétés sont utilisées pour configurer le tWaitForSocket s’exécutant dans le framework de Jobs Standard.

Le composant tWaitForSocket Standard appartient à la famille Orchestration.

Le composant de ce framework est toujours disponible.

### Basic settings

<table>
<thead>
<tr>
<th>Port</th>
<th>Saisissez le numéro du port à écouter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>End of line separator</td>
<td>Saisissez le séparateur de fin de ligne de vos données.</td>
</tr>
<tr>
<td>Then</td>
<td>Sélectionnez l’action à effectuer : soit continuer à écouter le port (<strong>keep on listening</strong>) soit fermer la connexion (<strong>close socket</strong>).</td>
</tr>
<tr>
<td>Print client/server data</td>
<td>Cochez cette case afin d’afficher les données client ou serveur.</td>
</tr>
</tbody>
</table>

### Advanced settings

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du composant. |

### Global Variables

| Global Variables | **ERROR_MESSAGE** : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option. **INPUT_DATA** : données transmises par le client. Cette variable est une variable **Flow** et retourne une chaîne de caractères. Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant. Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser. |

### Utilisation

#### Règle d’utilisation

Ce composant est un composant de début (ou de déclenchement) pour le sous-job exécuté en fonction de la condition établie. Ainsi, ce composant doit être connecté à un sous-job via un lien Iterate.

#### Connections

Liens de sortie (de ce composant à un autre) :
- **Row** : Iterate.

Liens d’entrée (d’un autre composant à celui-ci) :
- **Row** : Iterate.

Pour plus d’informations concernant les liens, consultez le *Guide utilisateur du Studio Talend*.

### Scénario associé

Aucun scénario n’est disponible pour la version Standard de ce composant.
tWaitForSqlData

Ce composant effectue une boucle sur une connexion donnée à la recherche d’un ajout ou d’une suppression de lignes, puis déclenche un sous-Job lorsque la condition relative à des données SQL est rencontrée.

**Propriétés du tWaitForSqlData Standard**

Ces propriétés sont utilisées pour configurer le tWaitForSqlData s’exécutant dans le framework de Jobs Standard.

Le composant tWaitForSqlData Standard appartient à la famille Orchestration.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Paramètre</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait at each iteration (in seconds)</td>
<td>Paramétrez en secondes l’intervalle de temps entre chaque vérification.</td>
</tr>
<tr>
<td>Max. iterations (infinite if empty)</td>
<td>Nombre de vérifications à effectuer avant que le Job ne s’arrête (si vous ne saisissez rien, la boucle sera infinie).</td>
</tr>
<tr>
<td>Use an existing connection/Component List</td>
<td>Cochez la case et cliquez sur le composant de connexion adéquat dans la liste Component list pour réutiliser les informations de connexion que vous avez déjà définies.</td>
</tr>
</tbody>
</table>

**Remarque :**

Lorsqu’un Job contient un Job parent et un Job enfant, la liste Component list présente uniquement les composants de connexion du Job du même niveau. Si vous souhaitez utiliser une connexion existant dans un autre niveau, vérifiez que les composants de connexion disponibles partagent la même connexion.

Pour plus d’informations concernant le partage d’une connexion à travers différents niveaux de Jobs, consultez Use or register a shared DB connection dans tous les composants de base de données que vous utilisez.

Sinon, vous pouvez également désactiver le composant de connexion et utiliser les Dynamic settings du composant, afin de paramétrer manuellement la connexion. Dans ce cas, vérifiez que le nom de la connexion est unique et distinct tout au long des deux niveaux de Jobs. Pour plus d’informations concernant les paramètres dynamiques (Dynamic settings), consultez le Guide utilisateur du Studio Talend.

<table>
<thead>
<tr>
<th>Table to scan</th>
<th>Nom de la table à vérifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger action when rowcount is</td>
<td>Sélectionnez la condition à rencontrer pour exécuter l’action :</td>
</tr>
</tbody>
</table>

| Equal to : égal à                                      |                                                                           |
Not Equal to : différent de  
Greater than : supérieur à  
Lower than : inférieur à  
Greater or equal to : supérieur ou égal à  
Lower or equal to : inférieur ou égal à.

<table>
<thead>
<tr>
<th>Value</th>
<th>Définissez la valeur à prendre en compte.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Then</td>
<td>Sélectionnez l’action à effectuer lorsque la condition est rencontrée : soit arrêter la boucle (exit loop), soit continuer la boucle jusqu’à ce que le nombre d’itération maximum soit atteint (continue loop).</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Global Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
<td></td>
</tr>
<tr>
<td>ROW_COUNT : nombre de lignes détectées dans la table. Cette variable est une variable Flow et retourne un entier.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Utilisation</th>
<th>Utilisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Règle d’utilisation</td>
<td>Bien que ce composant ait besoin d’un composant de connexion pour ouvrir l’accès à une base de données, il joue aussi le rôle de composant de début (ou de déclenchement) pour le sous-Job à exécuter lorsque la condition est rencontrée. Ainsi, ce composant doit être connecté à un sous-Job via un lien Iterate.</td>
</tr>
</tbody>
</table>

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.
**Scénario : Attendre l’ajout de lignes dans une table**

Ce scénario décrit un Job lisant la table d’une base de données dans l’attente d’un ajout de données dans cette table, pour ensuite exécuter un sous-Job. Lorsque des données sont ajoutées, le sous-Job effectue un Select* sur la table et affiche le contenu des données insérées dans la console standard.

- Cliquez et déposez les composants suivants de la Palette dans l’espace de modélisation : tMysqlConnection, tWaitForSqlData, tMysqlInput, tLogRow.
- Connectez le composant tMysqlConnection au composant tWaitForSqlData à l’aide un lien OnSubjobOK, disponible via le clic-droit.
- Puis connectez le composant tWaitForSqlData au sous-Job à l’aide d’un lien Iterate puisqu’aucune donnée n’est transférée pour le moment. En effet, le tWaitForSqlData effectue seulement des boucles jusqu’à ce que la condition soit rencontrée.
- Dans le sous-Job à exécuter si la condition est rencontrée, le composant tMysqlInput est connecté au tLogRow. Puisque cette connexion sert à transférer des données, utilisez un lien de type Row main.
- Maintenant, paramétrez la connexion à la table devant être vérifiée à intervalles réguliers. Dans l’onglet Basic settings de la vue Component du composant tMysqlConnection, paramétrez les propriétés de connexion à la base de données.

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Built-In</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>'localhost'</td>
</tr>
<tr>
<td>Port</td>
<td>'3306'</td>
</tr>
<tr>
<td>Database</td>
<td>'mytalsnndbb'</td>
</tr>
<tr>
<td>Username</td>
<td>'root'</td>
</tr>
<tr>
<td>Password</td>
<td>''</td>
</tr>
<tr>
<td>Encoding Type</td>
<td>ISO-8859-15</td>
</tr>
</tbody>
</table>

- Renseignez les champs Host, Port, Database, Username, Password pour ouvrir une connexion à la base de données. Pour plus d’informations, consultez les composants de connexion aux bases de données. Par exemple, pour une base de données Mysql, consultez tMysqlConnection à la page 2618.
- Dans le champ Encoding, sélectionnez l’encodage adéquat, si nécessaire.
- Puis sélectionnez le composant tWaitForSqlData, et dans l’onglet Basic settings de la vue Component, paramétrez ses propriétés.
- Dans le champ Wait at each iteration, paramétrez en secondes l’intervalle entre chaque itération.
Dans le champ **Max iterations**, saisissez le nombre d’itérations maximum à effectuer avant que le Job ne s’arrête.

Pour utiliser le composant **tWaitForSqlData**, il faut qu’une connexion soit ouverte pour effectuer le nombre de boucles défini. Sélectionnez la connexion adéquate (s’il y en a plusieurs) dans la liste déroulante **Component List**.

Dans le champ **Table to scan**, saisissez le nom de la table à scanner. Dans cet exemple : `test_datatypes`.

Dans les champs **Trigger action when rowcount is** et **Value**, sélectionnez la condition à rencontrer afin de lancer le sous-Job. Pour cet exemple, le nombre de lignes à prendre en compte dans la table scannée soit être supérieur ou égal à 1, ainsi sélectionnez **be greater or equal to 1**.

Dans le champ **Then**, sélectionnez l’action à effectuer lorsque la condition est rencontrée avant que le nombre maximum d’itérations défini ne soit atteint. Dans cet exemple, dès que la condition est rencontrée, la boucle devrait s’arrêter.

Puis paramétrez le sous-Job à exécuter lorsque la condition est rencontrée. Dans cet exemple, le sous-Job sélectionne les données de la table scannée et les affiche dans le console.

 Sélectionnez le composant **tMysqlInput**, et dans l’onglet **Basic settings** de la vue **Component**, paramétrez la connexion à la table.

Si la connexion est stockée dans le **Repository**, sélectionnez-la dans la liste déroulante. Sinon, cochez la case **Use an existing connection** et sélectionnez le composant de connexion correspondant dans la liste.

Dans cet exemple, le schéma correspondant à la structure de la table est stocké dans le **Repository**.

Dans le champ **Table Name**, renseignez le nom de la table à partir de laquelle les données seront extraites, `Test_datatypes`.
• Puis dans le champ **Query**, saisissez la commande `SELECT` permettant d’extraire le contenu de la table.

• Pour cet exemple, laissez les propriétés par défaut du composant **tLogRow**.

Avant d’exécuter le Job, assurez-vous que la table à scanner (**test_datatypes**) est bien vide, afin que la condition (**greater or equal to 1**) soit rencontrée. Puis exécutez le Job via le raccourci **F6**. Avant la fin de la boucle, ajoutez une ou deux lignes à la table **test_datatypes** afin que la condition soit rencontrée.

Le Job s’arrête lorsque les données ajoutées à la table sont détectées lors de la boucle et le contenu de la table est affiché dans la console.

``` plaintext
Starting job tWaitForSqlData at 16:55 06/03/2008
2| 143.112.32.4 -- [04/Mar/2008 00:00:00 +0100] "GET
/components/none HTTP/1.1" 404 354 "-" "Mozilla/4.0 (compatible
1| 143.112.32.4 -- [04/Mar/2008 00:00:00 +0100] "GET
/components/none HTTP/1.1" 404 354 "-" "Mozilla/4.0 (compatible
Job tWaitForSqlData ended at 16:55 06/03/2008. [exit code=0]
```
tWarn

Ce composant déclenche un avertissement souvent pris comme données de log exhaustives par le composant tLogCatcher.

Les composants tDie et tWarn sont étroitement liés au composant tLogCatcher. Ils sont généralement utilisés ensemble afin que les données de log collectées par le tLogCatcher soient rassemblées et envoyées vers la sortie définie.

Le tWarn envoie un message d'avertissement au composant suivant. En cas d'erreur, il n'arrête pas l'exécution du Job. Si vous souhaitez arrêter le Job en cours en cas d'erreur, consultez tDie à la page 674.

Propriétés du tWarn Standard

Ces propriétés sont utilisées pour configurer le tWarn s'exécutant dans le framework de Jobs Standard.

Le composant tWarn Standard appartient à la famille Logs & Errors.

Le composant de ce framework est toujours disponible.

**Basic settings**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Warn message</td>
<td>Saisissez votre message d'avertissement.</td>
</tr>
<tr>
<td>Code</td>
<td>Saisissez le niveau de code.</td>
</tr>
<tr>
<td>Priority</td>
<td>Sélectionnez le niveau de priorité.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>tStatCatcher Statistics</td>
<td>Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu'au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>WARN_MESSAGES : message d’avertissement. Cette variable est une variable After et retourne une chaîne de caractères.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WARN_CODE : niveau de code du message d’alerte. Cette variable est une variable After et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td>WARN_PRIORITY : niveau de priorité du message d’alerte. Cette variable est une variable After et retourne un nombre entier.</td>
</tr>
<tr>
<td></td>
<td>ERROR_MESSAGE : message d’erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
</tbody>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**.

### Utilisation

| Règle d’utilisation | Ce composant ne peut être utilisé comme composant de début. Ainsi, s’il est connecté à un composant de sortie, il doit aussi être connecté à un composant d’entrée. |

### Scénarios associés

Pour des exemples d’utilisation du **tWarn**, consultez les scénarios du composant **tLogCatcher** :

- Capturer les messages déclenchés par un composant **tWarn** à la page 2098.
- Capturer le message déclenché par un composant **tDie** à la page 2101.
tWebService

Ce composant invoque une méthode via un service Web afin de récupérer les valeurs des paramètres sélectionnés dans l’éditeur.

⚠️ **Avertissement** :

Ce composant requiert l’utilisation d’une JDK Oracle.

Le tWebService appelle la méthode du service Web invoqué et retourne la classe définie en fonction des paramètres donnés.

**Propriétés du tWebService Standard**

Ces propriétés sont utilisées pour configurer le tWebService s’exécutant dans le framework de Jobs Standard.

Le composant tWebService Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong></td>
<td>Propriétés utilisées ponctuellement.</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées.</td>
</tr>
</tbody>
</table>

**Service configuration**

Cliquez sur le bouton [...] pour ouvrir l’éditeur du composant. Dans l’éditeur, vous pouvez :

- sélectionner le service Web que vous souhaitez utiliser,
- configurer les paramètres d’entrée du service Web,
- configurer les paramètres de sortie du service Web permettant de récupérer les données à transmettre en sortie.

**Mapping links display as**

Auto : par défaut, les liens faisant la correspondance entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de courbes.

Curves : les liens faisant la correspondance entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de courbes.

Lines : les liens faisant la correspondance entre les schémas d’entrée et de sortie et les paramètres du service Web sont en forme de lignes droites. Cette dernière option améliore légèrement les performances.

**Connection Time out**

Saisissez en secondes la durée de connexion au service Web.
<table>
<thead>
<tr>
<th><strong>Receive Time out</strong></th>
<th>Saisissez en secondes le temps de réponse du serveur.</th>
</tr>
</thead>
</table>
| **Input schema**     | Un schéma est une description de lignes, il définit le nombre de champ qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Ce champ est utilisé pour traiter le schéma d’entrée. Les options pour ce schéma sont :  
  - **Built-in**: Propriétés utilisées ponctuellement.  
  - **Repository**: Sélectionnez le fichier de propriétés du composant. Les champs suivants sont alors pré-remplis à l’aide des données collectées. |
| **Edit Schema**      | Cliquez sur **Edit schema** pour modifier le schéma. Si le schéma est en mode **Repository**, trois options sont disponibles :  
  - **View schema**: sélectionnez cette option afin de voir le schéma.  
  - **Change to built-in property**: sélectionnez cette option pour passer le schéma en mode **Built-In** et effectuer des modifications locales.  
  - **Update repository connection**: sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur **No** et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| **Sync columns**     | Ce bouton est disponible lorsqu’un lien d’entrée est créé. Cliquez sur ce bouton pour récupérer le schéma du composant précédent. |
| **Output schema**    | Ce champ est utilisé pour traiter le schéma de sortie. Le schéma est soit built-in, soit distant dans le Repository. Il est configuré comme le schéma d’entrée.  
  | **Avertissement** :  
  Le schéma d’entrée n'est pas nécessairement identique à celui du schéma de sortie. |
| **Use NTLM**         | Cochez cette case si vous utilisez un protocole d'authentification NTLM.  
  **Domain** : Nom de domaine du client,  
  **Host** : Adresse IP du client. |
| **Need authentication** | Cochez la case d'authentification et renseignez le nom de l'utilisateur dans le champ **Username** et son mot de passe dans le champ **Password**, si cela est nécessaire pour accéder au service.  
  Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s'ouvre, saisissez le mot de passe entre guillemets |
| **Use http proxy** | Cochez cette case si vous vous connectez derrière un proxy et renseignez les informations correspondantes. |
| **Trust server with SSL** | Cochez cette case pour authentifier le serveur auprès du client via un protocole SSL et renseignez les champs correspondants.  
**TrustStore file** : saisissez le chemin d'accès et le nom du fichier TrustStore contenant la liste des certificats approuvés par le client.  
**TrustStore password** : saisissez le mot de passe utilisé pour vérifier l'intégrité des données TrustStore. |
| **Die on error** | Décochez cette case pour passer les lignes en erreur et terminer le traitement des lignes sans erreur. |

**Advanced settings**

| **Temporary folder (for wsdl2java)** | Définissez (ou parcourrez votre répertoire jusqu'à) un dossier temporaire configuré où stocker les fichiers wsdl. |
| **tStatCatcher Statistics** | Cochez cette case pour collecter les données de log au niveau du composant. |

**Global Variables**

| **Global Variables** | **ERROR_MESSAGE** : message d'erreur généré par le composant lorsqu'une erreur survient. Cette variable est une variable **After** et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.  
**NB_LINE** : nombre de lignes traitées. Cette variable est une variable **After** et retourne un entier.  
Une variable **Flow** fonctionne durant l'exécution d'un composant. Une variable **After** fonctionne après l'exécution d'un composant.  
Pour renseigner un champ ou une expression à l'aide d'une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
Pour plus d'informations concernant les variables, consultez le **Guide utilisateur du Studio Talend**. |

**Utilisation**

| **Règle d'utilisation** | Ce composant peut être utilisé comme composant intermédiaire ou comme composant de début de Job. Il requiert un composant de sortie. |
Scénario : Obtenir des noms de pays à l’aide d’un service Web

Ce scénario décrit un Job à trois composants utilisant un service Web pour récupérer le nom du pays correspondant à un code de pays donné.

Relier les composants

Procédure

1. Déposez un tFixedFlowInput, un tWebService et un tLogRow de la Palette dans l’espace de modélisation graphique.

2. Reliez les composants à l’aide de liens de type Main Row.

Configurer les données d’entrée

Procédure

1. Double-cliquez sur le composant tFixedFlowInput afin d’afficher sa vue Basic settings et définir ses propriétés.

2. Cliquez sur le bouton [...] à côté du champ Edit schema pour définir le schéma du composant d’entrée.

3. Dans la boîte de dialogue, cliquez sur le bouton [+] pour ajouter une colonne au schéma.

5. Cliquez sur **OK** pour fermer la boîte de dialogue. La colonne *CountryCode* s’affiche dans la table **Values** de la vue **Basic settings** du composant.

6. Dans le tableau **Values**, cliquez sur la colonne **Value** et saisissez la valeur d’entrée dans la colonne **CountryCode**, *cn* dans cet exemple. Ce code de pays est passé au **tWebService** afin de récupérer le nom du pays correspondant.

**Configurer le service Web**

** Sélectionnez le WSDL **

**Procédure**

1. Double-cliquez sur le composant **tWebService** pour ouvrir l’éditeur du composant. Vous pouvez également sélectionner le composant dans l’espace de modélisation graphique et, dans la vue **Basic settings**, cliquez sur le bouton [..] à côté du champ **Service configuration**.


3. Cliquez sur le bouton **Refresh** pour récupérer la description du WSDL dans les champs suivants.

4. Dans la liste **Port Name**, sélectionnez le port que vous souhaitez utiliser. Ici, sélectionnez le port `countrySoap`. 

**Configurer le mapping d’entrée**

**Procédure**

1. Cliquez sur **Next** pour passer à l’étape suivante.
2. Dans le panneau droit de la vue **Input mapping**, sélectionnez l’élément `parameters` de la zone de droite, en cliquant sur le bouton `+`, et en sélectionnant les paramètres souhaités dans la boîte de dialogue [Parameter Tree].
   
   Le service Web utilisé dans ce scénario ne comporte qu’un seul paramètre d’entrée : `CountryCode`.
   
   **Remarque :**
   
   S’il est disponible, utilisez le bouton **Auto map!** situé en bas à gauche de la fenêtre, il permet d’effectuer l’opération de mapping automatiquement.
   
   Vous devez créer une connexion entre le schéma d’entrée et les paramètres d’entrée du Service Web défini.
3. Dans la liste **Column**, sélectionnez la colonne du schéma d’entrée que vous souhaitez mettre en correspondance avec le paramètre de sortie du service Web et glissez-la dans le paramètre correspondant dans la zone de droite.

**Configurer le mapping de sortie**

**Procédure**

1. Cliquez sur **Next** pour ouvrir une nouvelle vue dans l’éditeur.
Dans la liste **Element** à gauche de l’éditeur, le paramètre de sortie du service Web apparaît automatiquement. Vous avez la possibilité d’ajouter d’autres paramètres en sélectionnant l’élément **[+] parameters** de la liste **Element** à gauche de l’éditeur, en cliquant sur le bouton **[+]**, et en sélectionnant les paramètres souhaités dans la boîte de dialogue **[Parameter Tree]**.

Le service Web utilisé dans ce scénario ne comporte qu’un seul paramètre de sortie : **GetCountryByCountryCodeResult**.

2. Dans la zone de droite, cliquez sur le bouton **[...]** du champ **Edit Schema** pour définir le schéma de sortie.

3. Dans la zone **Output** à droite de la boîte de dialogue, cliquez sur le bouton **[+]** pour ajouter une colonne au schéma de sortie.

5. Cliquez sur OK pour valider les changements et fermer la boîte de dialogue de définition de schéma.
   Vous devez à présent créer une connexion entre le paramètre de sortie du service Web défini et le schéma du composant de sortie.

6. Dans la liste Element à droite de l'éditeur, sélectionnez l'élément parameters.GetCountryByCountyCodeResult à gauche et glissez-le dans le champ correspondant à la colonne Result dans la zone de droite.

![Image de la fenêtre de l'éditeur de mapping]

**Remarque:**
S'il est disponible, utilisez le bouton Auto map! situé en bas à gauche de la fenêtre, il permet d'effectuer l'opération de mapping automatiquement.

7. Cliquez sur OK pour valider les changements et fermer l'éditeur.


9. Double-cliquez sur le tLogRow afin d'ouvrir sa vue Basic settings et cliquez sur Sync columns afin de récupérer le schéma du composant précédent.
**Exécuter le Job**

**Procédure**

Enregistrez votre Job et appuyez sur **F6** pour l'exécuter.

```
[statistics] connecting to socket on port 4077
[statistics] connected
<NewDataSet>
  <Table>
    <countrycode>cn</countrycode>
    <name>China</name>
  </Table>
  <Table>
    <countrycode>cn</countrycode>
    <name>China</name>
  </Table>
</NewDataSet>
[statistics] disconnected
```

**Résultats**

Le nom du pays correspondant au code de pays défini est retourné par le service Web.
Ce composant invoque une méthode via un service Web.
Le tWebServiceInput appelle la méthode du service Web invoqué et retourne la classe définie en fonction des paramètres donnés.

**Propriétés du tWebServiceInput Standard**

Ces propriétés sont utilisées pour configurer le tWebServiceInput s'exécutant dans le framework de Jobs Standard.
Le composant tWebServiceInput Standard appartient à la famille Internet.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Property type</th>
<th>Peut être Built-in ou Repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in : Propriétés utilisées ponctuellement.</td>
<td></td>
</tr>
<tr>
<td>Repository : Sélectionnez le fichier dans lequel sont stockées les propriétés du composant. Les champs suivants sont alors pré-remplis à l'aide des données collectées.</td>
<td></td>
</tr>
</tbody>
</table>

Cliquez sur cette icône pour ouvrir l'assistant de création de schéma WSDL et enregistrer les paramètres de connexion WSDL que vous avez définis dans la vue Basic settings du composant dans le Repository.

Pour plus d’informations sur la configuration et le stockage des paramètres de connexion WSDL, consultez le Guide utilisateur du Studio Talend.

**Schema et Edit schema**

Un schéma est une description de lignes, il définit le nombre de champ qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository.

Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la
<table>
<thead>
<tr>
<th><strong>métadonnée du schéma dans la fenêtre Repository Content.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in</strong> : Le schéma est créé et conservé pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.</td>
</tr>
<tr>
<td><strong>WSDL</strong></td>
</tr>
<tr>
<td>Description des bindings et de la configuration du service Web.</td>
</tr>
<tr>
<td><strong>Need authentication / Username et Password</strong></td>
</tr>
</tbody>
</table>
| Cochez cette case d’authentification et :
  - Saisissez un identifiant, ainsi qu’un mot de passe, dans les champs correspondants, si nécessaire pour accéder au service. Sinon,
  - Cochez la case **Windows authentication** et saisissez le nom de domaine Windows dans le champ correspondant si cela est nécessaire pour accéder au service.
  Pour saisir le mot de passe, cliquez sur le bouton [...] à côté du champ **Password**, puis, dans la boîte de dialogue qui s’ouvre, saisissez le mot de passe entre guillemets doubles et cliquez sur **OK** afin de sauvegarder les paramètres. |
| **Use http proxy** |
| Cochez cette case si vous vous connectez derrière un proxy et renseignez les informations correspondantes. |
| **Trust server with SSL** |
| Cochez cette case pour authentifier le serveur auprès du client via un protocole SSL et renseignez les champs correspondants.
  **TrustStore file** : saisissez le chemin d’accès et le nom du fichier TrustStore contenant la liste des certificats approuvés par le client.
  **TrustStore password** : saisissez le mot de passe utilisé pour vérifier l’intégrité des données TrustStore. |
| **Time out (second)** |
| Saisissez en secondes la durée de connexion au service Web. |
| **Method Name** |
| Saisissez le nom exact de la méthode à invoquer.
Le nom de la méthode **DOIT** correspondre à la méthode décrite dans le service Web. Le nom de la méthode est sensible à la casse. |
| **Parameters** |
| Saisissez les paramètres attendus et les valeurs à retourner. Assurez-vous que les paramètres saisis correspondent parfaitement aux noms et à la casse des paramètres décrits dans la méthode. |
Advanced settings

| Advanced Use | Cochez cette case pour afficher les champs suivants permettant d’utiliser les fonctions avancées du tWebServiceInput :
| WSDL2Java : cliquez sur le bouton [...] pour générer des routines contenant les codes Java nécessaires à la connexion et à l’interrogation du service Web.
| Code : saisissez le code permettant de se connecter et d’interroger le service Web en vous basant sur les deux modèles proposés en utilisant les routines générées automatiquement.
| Match Brackets : sélectionnez le nombre d’accolades à utiliser pour fermer la boucle for en fonction du nombre d’accolades ouvertes. |

| tStatCatcher Statistics | Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant. |

Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.  
| NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.  
| Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.  
| Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
| Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend. |

Utilisation

| Règle d’utilisation | Ce composant est généralement utilisé comme composant de début. Il requiert un composant de sortie. |

Scénario : Obtenir des noms de pays à l’aide d’un Webservice

Ce scénario décrit un Job constitué de deux composants dont le but est d’utiliser une méthode de service Web afin d’obtenir le nom du pays correspondant à un code de pays donné et d’afficher le résultat en sortie dans la console Run.
Procédure

1. Déposez un tWebServiceInput et un tLogRow depuis la Palette dans l’espace de modélisation graphique.
2. Reliez les deux composants à l’aide d’un lien Row > Main.

3. Double-cliquez sur le tWebServiceInput afin d’ouvrir sa vue Basic settings.


Dans cet exemple, le schéma est composé d’une seule colonne, Result.

6. Si vous vous connectez derrière un proxy, cochez la case **Use http proxy** et renseignez les champs concernant l’hôte, l’utilisateur, le port et le mot de passe.

7. Dans le champ **Method name**, saisissez le nom de la méthode à utiliser. Notez que le nom de la méthode est sensible à la casse.
   Dans cet exemple, le méthode `GetCountryByCountryCode` est utilisée.

8. Dans la zone **Parameters**, cliquez sur le bouton `[+]` pour ajouter une ligne au tableau, puis saisissez un code de pays, `fr` dans cet exemple.


![Execution](image)

**Résultats**

L’information sur le pays est retournée depuis le service Web et affichée dans la console **Run**.
tWriteJSONField

Ce composant transforme les données entrantes en champs JSON et les transfère dans un fichier, une table de base de données.

Configurer une arborescence JSON

Lorsque vous configurez une arborescence JSON, le type d’élément par défaut est *string*. Si un élément n’est pas de type *string*, vous devez ajouter un attribut à l’élément afin de définir son type.

- Pour un élément de type *integer, double, float* ou *boolean*, vous devez ajouter un attribut nommé *type* et définir sa valeur, dans la colonne **Static Value**, en *integer, number, float*, ou *boolean* respectivement.
- Pour un élément de type *array*, vous devez ajouter un attribut nommé *class* et définir sa valeur, dans la colonne **Static Value**, en *array* puis ajouter un sous élément nommé *element* défini en élément de boucle (*loop element*).
- Pour un élément de type *object*, vous devez ajouter un attribut nommé *class* et définir sa valeur, dans la colonne **Static Value**, en *object*.

L’image suivante montre un exemple de configuration d’arborescence JSON :

![Image montrant une configuration d’arborescence JSON](image-url)
**Propriétés du tWriteJSONField Standard**

Ces propriétés sont utilisées pour configurer le tWriteJSONField s’exécutant dans le framework de Jobs Standard.

Le composant tWriteJSONField Standard appartient à la famille Processing.

Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Output Column</th>
<th>Liste des colonnes définies dans le schéma de sortie contenant les champs JSON générés.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure JSON Tree</td>
<td>Cliquez pour ouvrir l’interface de création de la structure de données JSON. Pour plus d’informations, consultez Configurer une arborescence JSON à la page 4287.</td>
</tr>
</tbody>
</table>
| Schema et Edit Schema                                                        | Un schéma est une description de lignes, il définit le nombre de champs (colonnes) qui sont traités et passés au composant suivant. Lorsque vous créez un Job Spark, évitez le mot réservé line lors du nommage des champs. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
   • View schema : sélectionnez cette option afin de voir le schéma.
   • Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
   • Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
| Sync columns                                                                | Cliquez sur ce bouton pour synchroniser le schéma du fichier de sortie et le schéma du fichier d’entrée. La fonction Sync s’affiche uniquement lorsque le lien Row est connecté au composant de sortie. |
| Group by                                                                    | Définissez l’ensemble d’agrégation, les colonnes que vous souhaitez utiliser pour regrouper les données. |

**Built-In** : Le schéma est créé et conservé ponctuellement pour ce composant seulement. Voir également le Guide utilisateur du Studio Talend.  

Avertissement :
Assurez-vous que les données à grouper sont un ordre séquentiel.

Remove root node
Cochez cette case pour supprimer le nœud racine du champ JSON généré.

Advanced settings

Quote all values
Cochez cette case pour entourer de guillemets doubles toutes les valeurs, y compris les nombres et les booléens, dans le champ JSON généré.
Décochez cette case pour entourer uniquement les chaînes de caractères de guillemets doubles, dans le champ JSON généré.

Cochez cette case pour supprimer le nœud racine du champ JSON généré.

Cochez cette case pour collecter les données de log au niveau du composant.

Variables globales

Global Variables
ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères.
Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.
NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable After et retourne un entier.
Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.
Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Règle d’utilisation
Précédé d’un composant d’entrée, ce composant met les données d’entrée dans un champ JSON.

Scénario : Ecrire des données plates dans des champs JSON
Dans ce scénario, les données plates sont mises dans des champs JSON via le composant tWriteJSONField.
Relier les composants

**Procédure**

1. Déposez les composants suivants de la Palette dans l’espace de modélisation graphique : un tFixedFlowInput, un tWriteJSONField et un tLogRow.
2. Reliez le tFixedFlowInput au tWriteJSONField à l’aide d’un lien Row > Main.
3. Reliez le tWriteJSONField au tLogRow à l’aide d’un lien Row > Main.


Configurer les composants

**Procédure**

1. Double-cliquez sur le tFixedFlowInput pour afficher sa vue Basic settings.

Cliquez sur le bouton [+] pour ajouter trois colonnes, respectivement nommée firstname, lastname et dept, de type String.
Cliquez sur OK pour fermer l'éditeur.

3. Sélectionnez l'option Use Inline Content et saisissez les données ci-dessous dans le champ Content :

Andrew;Wallace;Doc  
John;Smith;R&D  
Christian;Dior;Sales

4. Cliquez sur le tWriteJSONField pour afficher sa vue Basic settings.

Cochez la case Remove root node afin de supprimer le nœud racine des champs JSON générés.

5. Cliquez sur le bouton [...] à côté du champ Configure JSON Tree pour ouvrir l'éditeur de l'arborescence XML.

Le schéma du tFixedFlowInput apparaît dans le panneau Linker source.

6. Dans le panneau Linker target, cliquez sur le nœud racine par défaut (rootTag) et saisissez staff, le nom du nœud racine du champ JSON à générer.

7. Cliquez-droit sur staff et sélectionnez Add Sub-element dans le menu contextuel.

8. Dans la boîte de dialogue, saisissez le nom du sous-nœud, firstname.
Répétez les étapes précédentes afin d’ajouter deux autres sous-nœuds, respectivement nommés `lastname` et `dept`.

9. Cliquez-droit sur `firstname` et sélectionnez `Set As Loop Element` dans le menu contextuel.
10. Déposez la ligne `firstname` du panneau `Linker source` dans la ligne du même nom, dans le panneau `Linker target`.

Dans la boîte de dialogue, sélectionnez `Add linker to target node`.

Cliquez sur `OK` pour fermer la boîte de dialogue.

Répétez ces étapes afin de lier les deux autres éléments.

Cliquez sur `OK` pour fermer l’éditeur d’arborescence XML.


12. Cliquez sur le bouton `[+]` du panneau de droite pour ajouter une colonne, nommée `staff`, qui contiendra les données JSON générées.

Cliquez sur `OK` pour fermer l’éditeur.

13. Double-cliquez sur le `tLogRow` pour afficher sa vue `Basic settings`. 
Sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

**Exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl + S** pour sauvegarder votre Job.
2. Cliquez sur **F6** pour l’exécuter.

```json
{|"first_name" : "Andrew", "last_name" : "Wallace", "dept" : "Doc"} |
{|"first_name" : "John", "last_name" : "Smith", "dept" : "R&D"} 
{|"first_name" : "Christian", "last_name" : "Dior", "dept" : "Sales"} |
```

Comme affiché ci-dessus, les champs JSON ont été correctement générés et les paramètres du nœud racine ont bien été supprimés.

**Scénarios associés**

Pour des scénarios associés, consultez

- **Scénario : Récupérer les messages d'erreur lors de l'extraction de données de champs JSON** à la page 1003.
- **Scénario : Extraire la structure d'un fichier XML et l'insérer dans les champs d'une base de données** à la page 4296.
- **Mapping de données XML** à la page 115.
tWriteXMLField

Ce composant lit un fichier XML d'entrée, extrait la structure du fichier et l’insère dans les champs définis du fichier XML de sortie.

Propriétés du tWriteXMLField Standard

Ces propriétés sont utilisées pour configurer le tWriteXMLField s’exécutant dans le framework de Jobs Standard.
Le composant tWriteXMLField Standard appartient à la famille XML.
Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Output Column</th>
<th>Sélectionnez la colonne du composant de sortie dans laquelle vous souhaitez écrire la structure XML.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure XML Tree</td>
<td>Ouvre l’interface d’aide à la création de la structure XML à écrire dans un champ. Pour plus d’informations sur l’interface, consultez Définir un arbre XML à la page 114.</td>
</tr>
</tbody>
</table>
| **Schema et Edit Schema** | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :

- **View schema** : sélectionnez cette option afin de voir le schéma.
- **Change to built-in property** : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.
- **Update repository connection** : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre [Repository Content]. |
<table>
<thead>
<tr>
<th><strong>Sync columns</strong></th>
<th>Cliquez sur ce bouton pour synchroniser le schéma de sortie avec celui d’entrée. La fonction <strong>Sync</strong> ne s’affiche que si une connexion de type Row est liée au composant de sortie.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group by</strong></td>
<td>Sélectionnez la colonne à utiliser pour regrouper les données.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th><strong>Remove the XML declaration</strong></th>
<th>Cochez cette case si vous ne souhaitez pas inclure la déclaration XML.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Create empty element if needed</strong></td>
<td>Cette case est cochée par défaut. Si le contenu de la colonne <strong>Related Column</strong> de l’éditeur d’arborescence XML est nul, ou si aucune colonne n’est associée au nœud XML, cette option créera une balise ouvrante et une balise fermante aux endroits prévus.</td>
</tr>
<tr>
<td><strong>Remarque</strong></td>
<td>Afin d’utiliser cette option, vous devez sélectionner le mode de génération <strong>Dom4J</strong>. Cette option est disponible lorsque la case <strong>Create empty element if needed</strong> est cochée.</td>
</tr>
<tr>
<td><strong>Create associated XSD file</strong></td>
<td>Si l’un des éléments XML est associé à un espace de nommage, cette option créera le fichier XSD correspondant.</td>
</tr>
<tr>
<td><strong>Remarque</strong></td>
<td>Pour utiliser cette option, vous devez sélectionner le mode de génération <strong>Dom4J</strong> dans la liste <strong>Generation mode</strong>.</td>
</tr>
<tr>
<td><strong>Advanced separator (for number)</strong></td>
<td>Cochez cette case pour modifier les séparateurs utilisés par défaut dans les nombres. <strong>Thousands separator</strong> : saisissez entre guillemets le séparateur à utiliser pour les milliers. <strong>Decimal separator</strong> : saisissez entre guillemets le séparateur à utiliser pour les décimales.</td>
</tr>
<tr>
<td><strong>Generation mode</strong></td>
<td>Sélectionnez le mode de génération correspondant à votre mémoire disponible. Les modes disponibles sont : • Lent et consommateur de mémoire (<strong>Slow and memory-consuming - Dom4J</strong>).</td>
</tr>
</tbody>
</table>
Remarque :
Cette option vous permet d'utiliser Dom4J pour traiter des fichiers XML très complexes.
- Rapide et peu consommateur de mémoire (*Fast with low memory consumption* (SAX)).

**Encoding**
Sélectionnez l’encodage à partir de la liste ou sélectionnez *Custom* et définissez-le manuellement. Ce champ est obligatoire pour la manipulation des données de base de données.

**tStatCatcher Statistics**
Cochez cette case pour collecter les données de log au niveau du composant.

### Global Variables

| Global Variables | ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case *Die on error* est décochée, si le composant a cette option.  
  
  NB_LINE : nombre de lignes lues par un composant d’entrée ou passées à un composant de sortie. Cette variable est une variable *After* et retourne un entier.  
  
  Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.  
  
  Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.  
  
  Pour plus d’informations concernant les variables, consultez le *Guide utilisateur du Studio Talend*. |

### Utilisation

| Règle d’utilisation | Ce composant doit être utilisé en composant intermédiaire. Il nécessite donc un composant d’entrée et de sortie. |

### Scénario : Extraire la structure d'un fichier XML et l’insérer dans les champs d’une base de données

Ce scénario est composé de trois composants et permet de lire un fichier XML, d’en extraire la structure puis de transmettre cette structure dans les champs d’une table.
**Procédure**

1. Glissez les composants `tFileInputXml` et `tWriteXMLField` de la famille `Xml` et un composant `tMysqlOutput` de la famille `Database > Mysql` de la Palette dans l'espace de modélisation graphique.

   Reliez ces composants via des liens de type Row > Main.

2. Double-cliquez sur le composant `tFileInputXml` pour paramétrer ses propriétés dans l'onglet Basic settings.

3. Dans la liste déroulante Property type, sélectionnez l'option Repository si la description de votre fichier est stockée dans une métadonnée du Repository. Si vous cliquez-déposez le composant directement à partir de la métadonnée, vous n'aurez pas besoin de modifier ses propriétés.

   Pour plus d'informations concernant le stockage des métadonnées dans la vue Repository consultez le Guide utilisateur du Studio Talend.

4. Sinon, sélectionnez l'option Built-in et renseignez manuellement les champs suivants. Pour plus d'informations sur les propriétés du composant `tFileInputXML`, consultez tFileInputXML à la page 1147.

   Si vous avez sélectionné l'option Built-in, cliquez sur le bouton [...] à côté du champ Edit schema puis dans la boîte de dialogue, décrivez manuellement la structure de votre fichier.


6. Cliquez sur le composant `tWriteXMLField` dans le Job designer puis cliquez sur la vue Component pour l'afficher et paramétrer les propriétés du composant :
7. Cliquez sur le bouton [...] à côté du champ Edit schema et, dans la boîte de dialogue, ajoutez une ligne en cliquant sur le bouton [+].

8. Dans cette ligne, saisissez le nom de la colonne de votre flux de sortie que vous souhaitez renseigner avec la structure du fichier XML dans la zone de droite, CustomerDetails dans ce scénario.

Dans la colonne Type de cette ligne, indiquez qu’elle est de type String et dans la colonne Length, indiquez qu’elle est de longueur 255.

Cliquez sur OK pour valider votre schéma de sortie et retourner aux propriétés du composant.

Dans le champ Output Column, sélectionnez la colonne dans laquelle vous souhaitez insérer le contenu XML.

9. Cliquez sur le bouton [...] du champ Configure Xml Tree pour ouvrir l’interface d’aide à la création de structures XML.
10. Dans la zone **Link Target**, cliquez sur le nœud *rootTag* et renommez-le *CustomerDetails*.

Dans la zone **Linker source**, sélectionnez *CustomerName* et *CustomerAddress* et glissez-les sur le nœud *CustomerDetails*. Une boîte de dialogue s'ouvre vous demandant quel type d'action vous souhaitez effectuer.

Sélectionnez **Create as sub-element of target node** afin de créer des sous-éléments au nœud *CustomerDetails*.

Cliquez-droit sur l’élément *CustomerName* et sélectionnez l’option **Set As Loop Element** dans le menu.

Cliquez sur **OK** pour valider la structure XML que vous avez définie.

11. Dans le Job designer, double-cliquez sur le composant **tMysqlOutput** pour paramétrer ses propriétés dans l’onglet **Basic settings**.

12. Si votre schéma est déjà stocké sous le nœud *Db Connections* dans le Repository, sélectionnez l’option **Repository** dans le champ **Schema** puis choisissez les métadonnées appropriées à partir de la liste.

Pour plus d’informations concernant le stockage des métadonnées dans la vue **Repository** consultez le Guide utilisateur du Studio Talend.

Si vous n’avez encore défini aucun schéma, sélectionnez l’option **Built-in** et renseignez manuellement les informations de connexion et la structure des données dans un schéma. Pour plus d’informations sur les propriétés du composant **tMysqlOutput**, consultez tMysqlSCD à la page 2712.

Dans le champ **Table**, saisissez le nom de la table dont les champs contiendront les données XML.
Dans le champ **Action on table**, sélectionnez l'opération que vous souhaitez effectuer sur la table. Pour ce scénario, sélectionnez **Create table** pour créer la table.

Dans le champ **Action on data**, sélectionnez l'opération que vous souhaitez effectuer sur les données. Pour ce scénario, laissez l'option **Insert**.

Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent. Vous pouvez cliquer sur le bouton [...] à côté du champ **Edit schema** pour consulter le schéma.

13. Enregistrez le Job et appuyez sur **F6** pour l'exécuter.

### Résultats

<table>
<thead>
<tr>
<th>CustomerDetails</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="CustomerDetails" /></td>
</tr>
</tbody>
</table>

Chaque champ de la colonne **CustomerDetails** est renseignée avec la structure XML du fichier de départ : l'instruction de traitement XML `<xml version="1.0" encoding="ISO-8859-15">`, le premier nœud séparant chaque client `<CustomerDetails>` et les informations sur les clients `<CustomerAddress>` et `<CustomerName>`.
tXMLMap

Ce composant transforme et route des données à partir d’une ou plusieurs source(s) de données vers une ou plusieurs destination(s).

Le tXMLMap est un composant avancé personnalisable permettant de transformer et de router des flux de données XML (données de type document), particulièrement pour traiter de nombreuses sources de données XML, avec ou sans jointure sur les données plates.

Propriétés du tXMLMap Standard

Ces propriétés sont utilisées pour configurer le tXMLMap s’exécutant dans le framework de Jobs Standard.

Le composant tXMLMap Standard appartient aux familles Processing et XML.

Le composant de ce framework est toujours disponible.

Basic settings

<table>
<thead>
<tr>
<th>Map Editor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Le Map Editor vous permet de définir les propriétés du routage et des transformations du tXMLMap.</td>
</tr>
</tbody>
</table>

Advanced settings

<table>
<thead>
<tr>
<th>tStatCatcher Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cochez cette case pour collecter les données de log au niveau du Job ainsi qu’au niveau de chaque composant.</td>
</tr>
</tbody>
</table>

Global Variables

<table>
<thead>
<tr>
<th>Global Variables</th>
</tr>
</thead>
</table>

Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plusieurs utilisations sont possibles, de la simple réorganisation des champs de données aux transformations les plus complexes, telles que le</td>
</tr>
</tbody>
</table>

4301
multiplexage et le démultiplexage de données, la concaténation, l'inversion, le filtrage, etc.

Lorsque c'est nécessaire, vous pouvez définir vos sorties pour les flux XML, à l'aide de l'élément "group", de l'élément "aggregate", d'éléments vides et de toute autre fonctionnalité, comme All in one. Pour plus d'informations concernant ces fonctionnalités, consultez le Guide utilisateur du Studio Talend.

Ce composant est utilisé comme composant intermédiaire et correspond parfaitement au processus nécessitant de nombreuses sources de données XML, comme les processus ESB de requête/réponse.

Limitation

Les limites sont les suivantes :
- L'utilisation du composant tXMLMap requiert des connaissances minimales en Java et XML afin d'exploiter au mieux ses fonctionnalités.
- Ce composant est une étape de jonction. Il ne peut donc être ni un composant d'entrée, ni un composant de sortie.
- Au moins un élément de boucle est requis pour chaque flux de données XML.

Les sections suivantes présentent plusieurs scénarios génériques concernant l'utilisation du composant tXMLMap. Si vous souhaitez consulter des exemples spécifiques d'utilisation de ce composant avec les composants ESB afin de construire des services de données, consultez les scénarios des composants suivants :

- Scénario 2 : Utiliser le tEBSConsumer avec des en-têtes SOAP personnalisés à la page 881.
- Scénario : Effectuer une requête sur les nom d'aéroports à partir des codes pays à la page 893.
- Scénario : Envoi d'un message via un service sans attente de réponse à la page 908.
- Scénario : Retourner une réponse "Hello world" à la page 919.

**Mapper et transformer des données de source XML**

Le scénario suivant crée un Job à trois composants pour mapper et transformer des données provenant d’un fichier source XML, *Customer.xml*, et générer un flux de sortie XML pouvant être réutilisé dans différents buts, comme, par exemple, pour une requête ESB basée sur la structure de l’arbre XML du fichier *Customer_State.xml*.

Ces trois composants sont :

- un tFileInputXML : utilisé pour fournir des données d’entrée au tXMLMap.
- un tXMLMap : mappe et transforme les flux de données XML en un seul flux de données XML.
- un tLogRow : utilisé pour afficher les données de sortie.
Le contenu du fichier XML *Customer.xml* se présente comme suit :

```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<Customers>
  <Customer RegisterTime="2001-01-17 06:26:40.000">
    <Name>
      <id>1</id>
      <CustomerName>Griffith Paving and Sealcoatin</CustomerName>
    </Name>
    <Address>
      <CustomerAddress>talend@apres91</CustomerAddress>
      <idState>2</idState>
    </Address>
    <Revenue>
      <Sum1>67852</Sum1>
    </Revenue>
  </Customer>
  <Customer RegisterTime="2002-06-07 09:40:00.000">
    <Name>
      <id>2</id>
      <CustomerName>Bill's Dive Shop</CustomerName>
    </Name>
    <Address>
      <CustomerAddress>511 Maple Ave. Apt. 1B</CustomerAddress>
      <idState>3</idState>
    </Address>
    <Revenue>
      <Sum1>88792</Sum1>
    </Revenue>
  </Customer>
  <Customer RegisterTime="1987-02-23 17:33:20.000">
    <Name>
      <id>3</id>
      <CustomerName>Glenn Oaks Office Supplies</CustomerName>
    </Name>
    <Address>
      <CustomerAddress>1859 Green Bay Rd.</CustomerAddress>
      <idState>2</idState>
    </Address>
    <Revenue>
      <Sum1>1225.</Sum1>
    </Revenue>
  </Customer>
  <Customer RegisterTime="1992-04-28 23:26:40.000">
    <Name>
      <id>4</id>
      <CustomerName>DBN Bank</CustomerName>
    </Name>
    <Address>
      <CustomerAddress>456 Grossman Ln.</CustomerAddress>
      <idState>3</idState>
    </Address>
    <Revenue>
      <Sum1>64493</Sum1>
    </Revenue>
  </Customer>
</Customers>
```
Le contenu du fichier XML *Customer_State.xml* se présente comme suit :

```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<customers>
  <customer id="1">
    <CustomerName>Griffith Paving and Sealcoating</CustomerName>
    <CustomerAddress>talend@apres91</CustomerAddress>
    <idState>2</idState>
  </customer>
  <customer id="2">
    <CustomerName>Bill's Dive Shop</CustomerName>
    <CustomerAddress>511 Maple Ave. Apt. 1B</CustomerAddress>
    <idState>3</idState>
  </customer>
</customers>
```

**Ajouter et relier les composants**

**Procédure**

1. Créez un nouveau Job et ajoutez un composant **tFileInputXML**, un **tXMLMap** et un **tLogRow**, en saisissant leur nom dans l'espace de modélisation graphique ou en les déposant depuis la **Palette**.

2. Renommez le **tFileInputXML** *Customers*, afin de mieux identifier son rôle.

   **Remarque :**

   Un composant utilisé dans l'espace de modélisation graphique peut être nommé comme vous le souhaitez. Pour plus d'informations concernant le renommage des composants, consultez le **Guide utilisateur du Studio Talend**.

3. Reliez le **tFileInputXML** nommé *Customers* au **tXMLMap** à l'aide d'un lien **Row > Main**.

4. Reliez le **tXMLMap** au **tLogRow** à l'aide d'un lien **Row > *New Output* (Main)**. Dans la boîte de dialogue, saisissez le nom du lien de sortie, *Customer* dans ce scénario.

**Configurer le flux d’entrée**

**Procédure**

1. Double-cliquez sur le **tFileInputXML** nommé *Customers* afin d’afficher sa vue **Basic settings**.
2. A côté du champ Edit schema, cliquez sur le bouton [...] afin d’ouvrir l’éditeur du schéma et, dans la boîte de dialogue [Schema], définissez le schéma en ajoutant une colonne Customer de type Document.

Notez que le type de données Document est essentiel pour tirer parti des fonctionnalités du tXMLMap. Pour plus d’informations concernant ce type de données, consultez le Guide utilisateur du Studio Talend.

3. Cliquez sur OK pour valider ce changement et fermer la boîte de dialogue. Une ligne est ajoutée automatiquement à la table Mapping.

4. Dans le champ File name/Stream, parcourez votre système jusqu’au fichier XML source fournissant les données client, ou bien saisissez son chemin d’accès entre guillemets doubles. Dans ce scénario, saisissez “E:/Customer.xml”.

5. Dans le champ Loop XPath query, saisissez une expression XPath entre guillemets doubles pour spécifier le nœud sur lequel la boucle est basée. Dans ce scénario, saisissez /, ce qui signifie effectuer une requête de recherche depuis la racine.

6. Dans la colonne XPath query de la table Mapping, saisissez entre guillemets doubles les champs à interroger. Dans ce scénario, saisissez ., ce qui signifie que tous les champs du nœud actuel (racine) seront extrait.

7. Cochez la case Get Nodes dans la colonne de la table Mapping.

Afin de construire le flux de données de type Document, il est nécessaire d’obtenir les nœuds de ce composant.
Configurer le tXMLMap pour effectuer des transformations

Procédure

1. Double-cliquez sur le composant tXMLMap pour ouvrir le Map Editor.

![Map Editor](image)

Notez que la zone d’entrée est déjà remplie par la structure XML simple par défaut et que la table supérieure est la table principale d’entrée (flux Main).

2. Dans la table d’entrée row1, cliquez-droit sur le nœud Customer et, dans son menu contextuel, sélectionnez Import From File. Dans la boîte de dialogue qui s’ouvre, parcourez votre système jusqu’au fichier XML source duquel importer la structure XML utilisée par les données reçues par le tXMLMap. Dans ce scénario, le fichier source XML est Customer.xml, le fichier des données d’entrée du composant tFileInputXML nommé Customers.

   Remarque :
   Vous pouvez également importer une arborescence XML à partir d’un fichier XSD. Lors de l’import d’une structure XML d’entrée ou de sortie depuis un fichier XSD, vous pouvez choisir un élément pour en faire la racine de votre structure XML. Pour plus d’informations concernant l’import d’une arborescence XML à partir d’un fichier XSD, consultez le Guide utilisateur du Studio Talend.

3. Dans la structure importée, cliquez-droit sur le nœud Customer et, dans le menu contextuel, sélectionnez As loop element pour le définir comme élément de boucle.
4. Dans la partie inférieure du Map Editor, cliquez sur l’onglet Schema editor afin d’afficher la vue correspondante. Du côté droit de cette vue, ajoutez une colonne Customer_States de type Document au schéma Customer. La racine XML correspondante est automatiquement ajoutée à la table de sortie Customer, en haut à droite, dans le flux de sortie.

5. Dans la table de sortie Customer, cliquez-droit sur le nœud Customer_States, et, dans le menu contextuel, sélectionnez Import From File. Dans la boîte de dialogue qui s’ouvre, parcourez votre système jusqu’au fichier XML duquel importer la structure XML. Dans ce scénario, le fichier est Customer_State.xml.

6. Cliquez-droit sur le nœud customer et sélectionnez As loop element dans le menu contextuel pour le définir comme élément de boucle.

7. Dans la table d’entrée row1, cliquez sur le nœud id et glissez-le dans la colonne Expression dans la ligne du nœud @id de la table de sortie Customer.
Répétez l’opération pour mapper CustomerName à CustomerName, CustomerAddress à CustomerAddress et idState à idState de la table d’entrée à la table de sortie.

**Remarque :**
Dans certains cas, vous pouvez avoir besoin de laisser des éléments vides dans votre arborescence XML de sortie. Dans ce cas, vous pouvez utiliser le tXMLMap pour les gérer. Pour plus d’informations concernant la gestion des éléments vides via le tXMLMap, consultez le Guide utilisateur du Studio Talend.

8. Cliquez sur la clé anglaise en haut de la table de sortie Customer et configurez la valeur de la propriété All in one à true. Pour plus d’informations concernant la fonctionnalité All in one, consultez le Guide utilisateur du Studio Talend.

9. Cliquez sur OK pour valider les modifications et fermer le Map Editor.

**Remarque :**
Si vous fermez le Map Editor sans avoir défini les éléments de boucle nécessaires, comme expliqué plus tôt dans ce scénario, l’élément racine est automatiquement défini comme l’élément de boucle.

**Configurer le tLogRow pour afficher les informations client**

**Procédure**
1. Double-cliquez sur le composant tLogRow pour ouvrir sa vue Basic settings.
2. Cliquez sur le bouton Sync columns afin de récupérer le schéma du composant précédent.
**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** pour sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.

Comme affiché ci-dessus, les informations clients transformées sont affichées dans la console.

**Créer un flux Lookup pour effectuer une jointure sur des données complémentaires**

Basé sur le scénario précédent, ce scénario explique comment utiliser un flux Lookup (de référence) afin d’effectuer une jointure sur des données souhaitées dans le fichier XML *USState.xml* du flux principal. Un autre **tFileInputXML** est ajouté au Job pour charger des données à partir du fichier lookup *USState.xml* vers le composant **tXMLMap** de la famille **Processing**.

Le contenu du fichier XML *USState.xml* se présente comme suit :

```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<USStates>
  <States>
    <idState>1</idState>
    <LabelState>Alabama</LabelState>
  </States>
  <States>
    <idState>2</idState>
    <LabelState>Connecticut</LabelState>
  </States>
</USStates>
```
Ajouter et relier un autre composant d’entrée

**Procédure**

1. Dans votre Studio, ouvrez le Job utilisé dans le scénario précédent pour l’afficher dans l’espace de modélisation graphique. Nommez ce composant *USStates* pour mieux identifier sa fonction.

2. Ajoutez un autre *tFileInputXML* au Job en saisisant son nom dans l’espace de modélisation graphique ou en le déposant depuis la Palette.

3. Reliez le *tFileInputXML* nommé *USStates* au *tXMLMap* à l’aide d’un lien *Row > Main*. La connexion est automatiquement modifiée en connexion Lookup.

Configurer le flux de référence

**Procédure**

1. Double-cliquez sur le composant *tFileInputXML* nommé *USStates* pour ouvrir sa vue *Basic settings*.

2. Cliquez sur le bouton [...] à côté du champ *Edit schema* et dans la boîte de dialogue [Schema], définissez le schéma en ajoutant une colonne *USState* de type *Document*. 
3. Cliquez sur OK pour valider les modifications et fermer la boîte de dialogue. Une ligne est automatiquement ajoutée à la table Mapping.

4. Dans le champ File name/Stream, parcourrez votre système ou saisissez entre guillemets doubles le chemin d'accès au fichier XML source contenant les données complémentaires. Dans ce scénario, le fichier est E:/USStates.xml.

5. Dans le champ Loop XPath query, saisissez une expression XPath entre guillemets doubles pour spécifier le nœud sur lequel se base la boucle. Dans ce scénario, saisissez /, ce qui signifie effectuer une requête de recherche depuis la racine.

6. Dans la colonne XPath query de la table Mapping, saisissez entre guillemets doubles les champs à interroger. Dans ce scénario, saisissez ., ce qui signifie que tous les champs du nœud (racine) sont extraits.


Configurer le tXMLMap pour la transformation

Procédure
1. Double-cliquez sur le composant tXMLMap pour ouvrir le Map Editor.
Notez que la zone d'entrée est déjà remplie par les tables d'entrée définies et que la table supérieure est la table principale d'entrée (flux Main).

2. Dans la table d'entrée row2, cliquez-droit sur le nœud USState et, dans le menu contextuel, sélectionnez Import From File. Dans la boîte de dialogue qui s'ouvre, parcourrez votre système jusqu'au fichier XML source duquel importer la structure XML utilisée par les données reçues par le tXMLMap. Dans ce scénario, le fichier XML source est USState.xml, le fichier des données d’entrée du tFileInputXML nommé USStates.

3. Dans l’arborescence XML importée, cliquez-droit sur le nœud States et, dans le menu contextuel, sélectionnez As loop element afin de le définir comme élément de boucle.
4. Dans la table principale d'entrée *row1*, cliquez sur le nœud *idState* et déposez-le, dans la colonne *Exp.key* de la ligne du nœud *idState*, dans la table lookup *row2*. Cela crée une jointure entre les deux tables d'entrée sur les données *idState*, parmi lesquelles le nœud *idState* du flux principal fournit la clé de référence (*lookup key*).

5. Dans la table lookup d'entrée *row2*, cliquez sur le nœud *LabelState* et déposez-le sur la table de sortie *Customer*. Une boîte de dialogue s'ouvre.

6. Dans cette boîte de dialogue, sélectionnez **Create as sub-element of target node** puis cliquez sur **OK**. Un sous-élément *LabelState* est ajouté dans l’arborescence XML de sortie et mappé avec le nœud *LabelState* de la table lookup.
7. Cliquez sur OK pour valider le mapping et fermer le Map Editor.

Sauvegarder et exécuter le Job

Procédure
1. Appuyez sur les touches Ctrl+S afin de sauvegarder votre Job.
2. Appuyez sur F6 pour exécuter le Job.

Comme affiché ci-dessus, les noms des États des États-Unis du fichier lookup dont les ID correspondent à ceux du fichier d’entrée principal sont ajoutés au flux de données et les informations combinées sont affichées dans la console.

Résultats
Mapper des données à l’aide d’un filtre

Ce scénario se base sur Créer un flux Lookup pour effectuer une jointure sur des données complémentaires à la page 4309 et explique comment appliquer une (des) condition(s) de filtre pour sélectionner les données qui vous intéressent, à l’aide du tXMLMap.

Mapper des données à l’aide d’un filtre

Procédure

1. Dans votre Studio, ouvrez le Job utilisé dans le scénario précédent afin de l’afficher dans l’espace de modélisation graphique.

2. Double-cliquez sur le tXMLMap pour ouvrir son éditeur Map Editor.

3. En haut de la table de sortie Customer, cliquez sur le bouton pour ouvrir la zone de filtre.


5. Cliquez sur OK pour valider les modifications et fermer l’éditeur.


Résultats

Comme affiché ci-dessus, les résultats montrent que les clients Griffith Paving and Sealcoating et Glenn Oaks Office Supplies dont l’ID de l’État est 2 sont affichés dans la console.

Capturer les données rejetées par le flux Lookup et le filtre

Les données rejetées par le flux Lookup et les conditions de filtre définies dans le tXMLMap peuvent être capturées et écrites en sortie par le composant lui-même.
Ce scénario se base sur Mapper des données à l'aide d'un filtre à la page 4315 et explique comment capturer les données rejetées par le flux Lookup et le filtre, définis dans les scénarios précédents. Dans ce scénario, un autre composant **tLogRow** est ajouté au Job du scénario précédent pour afficher les données rejetées.

**Ajouter et relier un composant de sortie supplémentaire**

**Procédure**

1. Dans le Studio, ouvrez le Job utilisé dans le scénario précédent dans l’espace de modélisation graphique.
2. Ajoutez un autre composant **tLogRow** au Job en saisissant son nom dans l’espace de modélisation graphique ou en le déposant depuis la Palette.
3. Reliez le **tXMLMap** au second **tLogRow**, à l’aide d’un lien **Row > "New Output" (Main)**. Une boîte de dialogue s’ouvre et vous demande de nommer ce lien de sortie. Dans ce scénario, nommez le lien **Reject**.
Configurer le tXMLMap pour la transformation

Procédure

1. Double-cliquez sur le composant tXMLMap pour ouvrir son éditeur. Une table *Reject* vide a été ajoutée à la sortie pour représenter le flux de sortie contenant les données rejetées.

2. Dans la table principale d’entrée *row1*, cliquez sur le nœud *id* et déposez-le dans la table *Reject*. Une colonne *id* est ajoutée au schéma *Reject* dans la zone *Schema editor* en bas de l’éditeur.

3. Déposez également *CustomerName*, *CustomerAddress* et *idState* de la table *row1* et *LabelState* de la table *row2* dans la table de sortie *Reject*. Quatre colonnes, *CustomerName*, *CustomerAddress*, *idState* et *LabelState* sont ajoutées au schéma *Reject* dans la zone *Schema editor*.

   **Remarque :**

4. En haut de la table *Reject* de sortie, cliquez sur le bouton pour ouvrir la zone de configuration.
5. Configurez la valeur de la propriété **Catch Output Reject** à **true** afin de récupérer les données rejetées par le filtre du scénario précédent, pour le flux de sortie **Customer**.

6. Configurez la valeur de la propriété **Catch Lookup Inner Join Reject** à **true** pour récupérer les données rejetées par l'opération de jointure Inner Join.

7. Cliquez sur **OK** pour valider les modifications et fermer l’éditeur de mapping.

**Configurer le flux de sortie**

**Procédure**

1. Double-cliquez sur le second composant **tLogRow** pour ouvrir sa vue **Basic settings**.
2. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.
3. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)** pour un affichage optimal des résultats.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job.
2. Appuyez sur **F6** pour exécuter le Job.

**Résultats**

Les données capturées rejetées par le flux **Lookup** et le filtre s'affichent comme suit dans la vue **Run**:
Comme affiché ci-dessus, les données dont la valeur de idState est 2 sont sélectionnées par le filtre configuré dans le scénario précédent et affichées dans la partie supérieure. Les données dont la valeur de idState n’est pas 2 sont rejetées et affichées dans la partie inférieure.

**Mapper des données à l’aide d’un élément “group”**

Ce scénario se base sur le Job utilisé dans Créer un flux Lookup pour effectuer une jointure sur des données complémentaires à la page 4309 et explique comment définir un élément en tant que group element dans le Map Editor du tXMLMap afin de regrouper les données de sortie. Pour plus d’informations concernant le regroupement de données de sortie à l’aide du composant tXMLMap, consultez le Guide utilisateur du Studio Talend.

L’objectif de ce scénario est de regrouper les IDs des clients et les informations relatives aux clients, selon l’Etat dans lequel ils résident. Vous devez reconstruire l’arborescence XML de la table de sortie Customer, en prenant en compte les facteurs suivants :

- L’élément de boucle et ses sous-éléments doivent dépendre directement de l’élément de groupe.
- L’élément relatif aux informations d’Etat utilisé comme condition de groupe doit dépendre directement de l’élément de groupe.
• L’élément de groupe ne peut pas être l’élément racine.

À partir de cette analyse, la structure XML des données de sortie doit se présenter comme suit. Le nœud customers est le nœud racine, le nœud customer est configuré comme l’élément de groupe et les données de sortie dont groupées selon l’élément LabelState.

Pour qu’un élément de groupe agisse, les données XML à traiter doivent avoir été triées, par exemple via vos outils XML, autour de l’élément qui sera utilisé comme condition de regroupement. Dans cet exemple, les clients ayant un même ID d’État doivent être regroupés. Les données d’entrée du fichier XML Customer.xml doivent se présenter comme suit :

```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<Customer>
  <RegisterTime>2001-01-17 06:26:40.000</RegisterTime>
  <Name id="1"><CustomerName>Griffith Paving and Sealcoating</CustomerName></Name>
  <Address idState="2"><CustomerAddress>talend@apres91</CustomerAddress></Address>
  <Revenue><Sum1>67852</Sum1></Revenue>
</Customer>
<Customer>
  <RegisterTime>1987-02-23 17:33:20.000</RegisterTime>
  <Name id="3"><CustomerName>Glenn Oaks Office Supplies</CustomerName></Name>
  <Address idState="2"><CustomerAddress>1859 Green Bay Rd.</CustomerAddress></Address>
  <Revenue><Sum1>1225</Sum1></Revenue>
</Customer>
<Customer>
  <RegisterTime>2002-06-07 09:40:00.000</RegisterTime>
  <Name id="2"><CustomerName>Bill's Dive Shop</CustomerName></Name>
  <Address idState="3"><CustomerAddress>511 Maple Ave. Apt. 1B</CustomerAddress></Address>
  <Revenue><Sum1>88792</Sum1></Revenue>
</Customer>
<Customer>
  <RegisterTime>1992-04-28 23:26:40.000</RegisterTime>
  <Name id="4"><CustomerName>Acme Bike Shop</CustomerName></Name>
  <Address idState="3"><CustomerAddress>1234 Oak St.</CustomerAddress></Address>
  <Revenue><Sum1>2500</Sum1></Revenue>
</Customer>
```
Mapper des données à l'aide d'un élément "group"

**Procédure**

1. Dans votre Studio, ouvrez le Job utilisé dans Créer un flux Lookup pour effectuer une jointure sur des données complémentaires à la page 4309 afin de l'afficher dans l'espace de modélisation graphique et double-cliquez sur le tXMLMap pour ouvrir le Map Editor.

2. Dans l'arborescence XML de la table de sortie Customer, cliquez-droit sur le nœud customer (loop) et sélectionnez Delete dans le menu contextuel. Tous les éléments sous le nœud racine customers sont supprimés. Vous pouvez reconstruire l'arborescence XML à utiliser pour grouper vos données.

![Map Editor](image)

3. Cliquez-droit sur le nœud racine customers et sélectionnez Create Sub-Element dans le menu contextuel. Dans la boîte de dialogue qui s'ouvre, saisissez le nom du sous-élément, customer, par exemple.

4. Cliquez sur OK pour valider les modifications et fermer la boîte de dialogue. Un nœud racine customer est ajouté à la table de sortie.

5. Dans la table d'entrée row2, sélectionnez le nœud LabelState et déposez-le sur le nœud customer de la table de sortie. Dans la boîte de dialogue qui s'ouvre, sélectionnez Create as sub-element of target node puis cliquez sur OK pour fermer la boîte de dialogue. Un nœud LabelState est ajouté sous le nœud customer dans la table de sortie.

6. Cliquez-droit sur le nœud customer de la table de sortie et sélectionnez Create Sub-Element dans le menu contextuel. Dans la boîte de dialogue qui s'ouvre, saisissez le nom du nouveau sous-élément. Dans cet exemple, saisissez Name.

7. Cliquez sur OK pour valider les modifications et fermer la boîte de dialogue. Un nœud Name est ajouté sous le nœud customer dans la table de sortie.
8. Dans la table d’entrée row1, sélectionnez les nœuds id et CustomerName et déposez-les sur le nœud Name de la table de sortie. Dans la boîte de dialogue qui s’ouvre, sélectionnez Create as sub-element of target node et cliquez sur OK pour fermer la boîte de dialogue. Un nœud id et un nœud CustomerName sont ajoutés sous le nœud Name de la table de sortie.

9. Dans la table de sortie, cliquez-droit sur le nœud Name et sélectionnez As loop element dans le menu contextuel pour définir cet élément comme élément de boucle. Cliquez-droit sur le nœud customer et, dans le menu contextuel, sélectionnez As group element afin de regrouper les données de sortie selon l’élément LabelState.

10. Cliquez sur OK pour valider les modifications et fermer l’éditeur de mapping.

11. Appuyez sur les touches Ctrl+S afin de sauvegarder votre Job puis sur F6 pour l’exécuter.

Résultats

Comme affiché ci-dessus, les éléments id et CustomerName contenus dans la boucle sont regroupés dans l’élément LabelState. La balise du group element customer marque le début et la fin de chaque groupe.

Classer les données de sortie avec l’élément "aggregate"

Ce scénario se base sur Mapper des données à l’aide d’un élément "group" à la page 4320, il explique comment définir un élément "aggregate element" dans le Map Editor du tXMLMap, afin de classer les données de sortie dans des flux XML séparés. Pour plus d’informations concernant l’agrégation de données de sortie à l’aide du tXMLMap, consultez le Guide utilisateur du Studio Talend.
L'objectif de ce scénario est de classer les informations relatives à l'ID d'un client et à son nom, à l'aide d'un élément *aggregate*, selon l'État d'où ils viennent puis d'envoyer ces données séparément dans différents flux XML vers le composant suivant.

Pour qu'un élément *aggregate* soit pris en compte, les données XML à traiter doivent avoir été triées, par exemple via vos outils XML, autour de l'élément à utiliser comme condition d'agrégation. Dans cet exemple, les clients ayant le même ID d'État doivent être groupés. Les données d'entrée dans le fichier XML *Customer.xml* doivent être les mêmes que dans *Mapper des données à l'aide d'un élément "group"* à la page 4320.

**Scénario 6 : Classer les données de sortie avec l’élément "aggregate"**

**Procédure**

1. Dans votre Studio, ouvrez le Job utilisé dans *Mapper des données à l'aide d'un élément "group"* à la page 4320 et double-cliquez sur le *tXMLMap* pour ouvrir son Map Editor.
2. Cliquez-droit sur l'élément *customer* dans la table de sortie afin d'ouvrir son menu contextuel et sélectionnez *Remove group element*.
3. En haut de la table de sortie, cliquez sur la clé anglaise afin de configurer la propriété *All in one* à *false*.

Afin de rendre l'élément *aggregate* disponible, vérifiez que la valeur de la propriété *All in one* est *false*. Pour plus d'informations concernant cette fonctionnalité, consultez le *Guide utilisateur du Studio Talend*.
5. Cliquez sur **OK** afin de valider ces modifications et fermer le **Map Editor**.

6. Appuyez sur les touches **Ctrl+S** pour sauvegarder votre Job puis sur **F6** pour l’exécuter.

**Résultats**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<customers>
<customer><LabelState>Connecticut</LabelState><Name><id>1</id><CustomerName>Griffith Paving and Sealcoating</CustomerName></Name><Name><id>3</id><CustomerName>Glenna Oaks Office Supplies</CustomerName></Name><customer></customer>
<customer><LabelState>Ohio</LabelState><Name><id>2</id><CustomerName>Bill's Dive Shop</CustomerName></Name><Name><id>4</id><CustomerName>DSW Bank</CustomerName></Name><customer></customer>
</customers>
```

Comme affiché ci-dessus, le **tXMLMap** écrit deux flux XML séparés, chacun d’entre eux contenant les informations d’un État et des clients vivant dans cet État.

**Scénario 7 : Restructurer des données produit à l’aide de différents éléments de boucle**

Ce scénario utilise un Job à quatre composants pour restructurer les données produit fournies par un fichier XML source, *ProductsIn.xml* à l’aide de différents éléments de boucle (**loop element**).

Les composants utilisés sont :

- un **tFileInputXML** : lit les données produit source et les passe au **tXMLMap**.
• un tXMLMap : transforme le flux d'entrée pour lui donner la structure simplifiée attendue.
• un tLogRow : affiche les résultats d'exécution dans la console.
• un tFileOutputXML : écrit un flux de sortie dans un fichier XML.

Le contenu du fichier XML source *ProductsIn.xml* se présente comme suit :

```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<products category="1" name="laptop">

<!-- Summary -->
<summary>
<company>DELL, HP</company>
<sales unit="Dollars">12345678910.12345</sales>
$model>business</model>
</summary>

<manufacture id="manu_1" date="2012-10-30">
 <name>DELL</name>
</manufacture>
<manufacture id="manu_2" date="2012-10-28">
 <name>HP</name>
</manufacture>

<types model="business1">
 <type>DELL123</type>
 <manufacture_id>manu_1</manufacture_id>
</types>
<types model="business2">
 <type>HP123</type>
 <manufacture_id>manu_2</manufacture_id>
</types>

<sales>
 <sale unit="Dollars" type="DELL123">
  <quater>1</quater>
  <income>12345</income>
 </sale>
 <sale unit="Dollars" type="HP123">
  <quater>1</quater>
  <income>12345.123</income>
 </sale>
</sales>
</products>
```

L'objectif de ce scénario est la restructuration de données produits est de simplifier la présentation des informations produit afin de réutiliser le fichier pour la manufacture. Les données attendues se présentent comme suit. L'élément racine est modifié à *manufacturers*, les informations relatives aux ventes sont consolidées dans l'élément *sale* et l'élément *manufacturer* est réduit à un niveau.

```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<manufacturers category="1" name="laptop">

<sales unit="Dollars">
 <sale sales_type="DELL123">12345.0</sale>
 <sale sales_type="HP123">12345.123</sale>
</sales>

<manufacturer id="manu_1" date="03-04-0036" name="DELL"/>
<manufacturer id="manu_2" date="04-04-0034" name="HP"/>
</manufacturers>
```

4326
Ajouter et relier les composants

Procédure

1. Créez un nouveau Job et ajoutez un tFileInputXML, un tXMLMap, un tLogRow et un tFileOutputXML de la Palette en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.
2. Reliez le tFileInputXML au tXMLMap à l’aide d’un lien Row > Main.
3. Reliez le tXMLMap au tLogRow à l’aide d’un lien Row > *New Output* (Main). Dans la boîte de dialogue qui s’ouvre, saisissez le nom de la connexion de sortie, outDoc dans cet exemple.
4. Reliez le tLogRow au tFileOutputXML à l’aide d’un lien Row > Main.

Configurer le flux d’entrée

Procédure

1. Double-cliquez sur le tFileInputXML pour ouvrir sa vue Basic settings.

2. Cliquez sur le bouton […] à côté du champ Edit schema et dans la boîte de dialogue [Schema], définissez le schéma en ajoutant une colonne doc de type Document.
3. Cliquez sur OK afin de valider les modifications et fermer la boîte de dialogue. Une ligne est ajoutée automatiquement à la table Mapping.

4. Dans le champ File name/Stream, parcourez votre système ou saisissez entre guillemets doubles le chemin d'accès au fichier source XML fournissant les données d'entrée produits. Dans ce scénario, saisissez E:/ProductsIn.xml.

5. Dans le champ Loop XPath query, saisissez une expression XPath entre guillemets doubles pour spécifier le nœud sur lequel se base la boucle. Dans ce scénario, saisissez /, ce qui signifie effectuer une requête de recherche depuis la racine.

6. Dans la colonne XPath query de la table Mapping, saisissez entre guillemets doubles les champ s à interroger. Dans ce scénario, saisissez ., ce qui signifie que tous les champs du nœud actuel (racine) seront extraits.

7. Dans la colonne Get Nodes de la table Mapping, cochez la case correspondante.

**Configurer le tXMLMap avec différentes boucles**

**Procédure**

1. Double-cliquez sur le composant tXMLMap pour ouvrir le Map Editor.
Notez que la zone d’entrée est déjà renseignée avec la structure XML simple par défaut et que la table du haut est la table d’entrée principale.

2. Dans la table d’entrée `row1`, cliquez-droit sur le nœud `doc` et, dans le menu contextuel, sélectionnez `Import From File`. Dans la boîte de dialogue qui s’ouvre, parcourez votre système jusqu’au fichier source XML pour importer la structure XML à utiliser par les données reçues par le tXMLMap. Dans ce scénario, le fichier source XML est `ProductsIn.xml` et il contient les données d’entrée du tFileInputXML.

3. Dans l’arborescence XML importée, cliquez-droit sur le nœud `manufacturer` et sélectionnez `As loop element` dans le menu contextuel afin de le définir comme élément de boucle. Répétez l’opération avec les nœuds `types` et `sale`, respectivement.

5. Dans la table `outDoc` de sortie, importez la structure XML des données à utiliser depuis le fichier XML contenant les données de sortie attendues et fournissez la structure XML attendue.

Cliquez-droit sur le nœud `sale` de la table de sortie et sélectionnez **As loop element** dans le menu contextuel. Répétez l’opération avec les nœuds `manufacturer` et `types`.

6. Dans la table d’entrée `row1`, cliquez sur le nœud `@category` et déposez-le dans le champ `Expression`, dans la ligne du nœud `@category` dans la table de sortie `outDoc`. 
Répétez l'opération afin de mapper les nœuds suivants :

- @name à @name.
- @unit sous le nœud summary à @unit.
- @id à @id et à manufacture id, respectivement.
- @date à @date.
- name à @name.
- type à type.
- @type à @sales_type.
- income à sale (loop).

7. En haut de la table outDoc de sortie, cliquez sur l'icône de clé anglaise et configurez la valeur de la propriété All in one à true afin de générer un flux XML. Pour plus d'informations concernant la fonctionnalité All in one, consultez le Guide utilisateur du Studio Talend.

8. Cliquez sur le bouton [...] à côté de l’élément de boucle manufacture et, dans la boîte de dialogue [Configure source loops] qui s’ouvre, cliquez une fois sur le bouton [+] pour ajouter une boucle source, manufacturer. Répétez l’opération pour ajouter une boucle source sale pour l’élément de boucle sale.
9. Cliquez sur le bouton [ ] à côté de l'élément boucle types et, dans la boîte de dialogue [Configure source loop], ajoutez deux boucles source, types et manufacturer. Assurez-vous que le numéro de séquence des types de la boucle source 0. Ainsi, la partie relative du flux de sortie sera triée selon les valeurs de l'élément type.

**Remarque :**

Lorsqu'un élément de boucle reçoit des mappings d'un ou plusieurs élément(s) de boucle du flux d'entrée, cela vous permet de configurer la séquence des boucles en entrée. Par exemple, dans ce scénario, l'élément de boucle types du flux de sortie est mappé avec le nœud @id, appartenant à l'élément de boucle manufacturer et avec le nœud type, appartenant à l'élément de boucle types du flux d'entrée. Le flux de sortie est trié selon la boucle primaire types.

10. Cliquez sur OK pour valider les mappings et fermer le Map Editor.

**Configurer le flux de sortie**

**Procédure**

1. Double-cliquez sur le tLogRow pour ouvrir sa vue Basic settings.

2. Cliquez sur le bouton Sync columns pour récupérer le schéma du composant précédent et acceptez la propagation proposée par la boîte de dialogue qui s'ouvre.
3. Double-cliquez sur le tFileOutputXML pour ouvrir sa vue Basic settings.

4. Dans le champ File Name, parcourez votre système ou saisissez le chemin d'accès au fichier dans lequel les données de sortie seront écrites. Dans ce scénario, le chemin d'accès est E:/ProductsOut.xml.

5. Cochez la case Incoming record is a document.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches Ctrl+S afin de sauvegarder le Job.
2. Appuyez sur F6 pour l'exécuter.
Comme affiché ci-dessus, les données produit d’entrée sont restructurées comme prévu et les données de sortie s’affichent dans la console et sont écrites dans le fichier XML `ProductsOut.xml`. 
tXMLRPCInput

Ce composant invoque une méthode via un service Web dont l’objectif est celui décrit dans les propriétés.

Ce composant appelle la méthode du service RPC invoqué et retourne la classe définie en fonction des paramètres donnés.

Propriétés du tXMLRPCInput Standard

Ces propriétés sont utilisées pour configurer le tXMLRPCInput s’exécutant dans le framework de Jobs Standard.

Le composant tXMLRPCInput Standard appartient à la famille Internet.

Le composant de ce framework est toujours disponible.

Basic settings

| Schema et Edit Schema | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant. Le schéma est soit local (built-in) soit distant dans le Repository. Cliquez sur Edit schema pour modifier le schéma. Si le schéma est en mode Repository, trois options sont disponibles :
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• View schema : sélectionnez cette option afin de voir le schéma.</td>
<td></td>
</tr>
<tr>
<td>• Change to built-in property : sélectionnez cette option pour passer le schéma en mode Built-In et effectuer des modifications locales.</td>
<td></td>
</tr>
<tr>
<td>• Update repository connection : sélectionnez cette option afin de modifier le schéma stocké dans le référentiel et décider de propager ou non les modifications à tous les Jobs. Si vous souhaitez propager les modifications uniquement au Job courant, cliquez sur No et sélectionnez à nouveau la métadonnée du schéma dans la fenêtre Repository Content.</td>
<td></td>
</tr>
<tr>
<td>Server URL</td>
<td>Adresse URL du service RPC à accéder</td>
</tr>
<tr>
<td>Need authentication / Username et Password</td>
<td>Cochez la case d’authentification et renseignez le nom de l’utilisateur (username) et son mot de passe (password), si cela est nécessaire pour accéder au service.</td>
</tr>
<tr>
<td>Method Name</td>
<td>Saisissez le nom exact de la méthode à invoquer.</td>
</tr>
</tbody>
</table>
Le nom de la méthode DOIT correspondre parfaitement à la méthode décrite dans le service RPC. Le nom de la méthode est sensible à la casse.

Return class

Sélectionnez le type de données à retourner par la méthode. Assurez-vous qu’il correspond parfaitement à celui défini dans la méthode.

Parameters

Saisissez les paramètres attendus en entrée par la méthode.

Advanced settings

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

Global Variables

ERROR_MESSAGE : message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.

NB_LINE : nombre de lignes traitées. Cette variable est une variable After et retourne un entier.

Une variable Flow fonctionne durant l’exécution d’un composant. Une variable After fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches Ctrl+Espace pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

Utilisation

Ce composant est généralement utilisé comme composant de début. Il nécessite un composant de sortie.

Scénario : Chercher le nom d’un État via une méthode XMLRPC

Ce scénario décrit un Job à deux composants utilisant une méthode RPC et affichant le résultat dans la console.
• Cliquez et déposez les composants **tXMLRPCInput** et **tLogRow**.

• Dans l’onglet **Basic settings**, paramétrez les propriétés du composant **tXMLRPCInput**.

  **Schema Type**<br>  Built-in<br>  Edit schema<br>

  **Server url**<br>  “http://phpxmlrpc.sourceforge.net/server.php”<br>

  **Method**<br>  “examples.getStateName”<br>  return class java.lang.String.class<br>

  **Parameters**<br>  

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>State Nr</td>
<td>42</td>
<td>java.lang.Byte.class</td>
</tr>
</tbody>
</table>

• Dans la liste déroulante **Schema type**, sélectionnez l’option **Built-in**.

• Configurez un schéma contenant une unique colonne puisque seul un paramètre est attendu en sortie : **StateName**.


• Aucune information d’authentification n’est nécessaire pour cet exemple.

• Dans le champ **Method**, renseignez la méthode à appeler, ici : **examples.getStateName**

• Pour cet exemple, la classe retournée n’est pas obligatoire. Ainsi, dans le champ **return class**, laissez les paramètres par défaut.

• Puis, dans le tableau **Parameters**, configurez les paramètres nécessaires à la méthode appelée. La colonne **Name** n’est pas utilisée dans le code mais la valeur doit correspondre à la syntaxe attendue par la méthode. Dans cet exemple, saisissez **State Nr** dans le champ **Name** et dans le champ **Value**, choisissez un numéro au hasard, ici : 42.

• La classe n’a pas beaucoup d’impact sur cette méthode démo mais pourrait en avoir sur une autre méthode. Ainsi, laissez les paramètres par défaut.

• Dans la vue **Component** du composant **tLogRow**, cochez la case **Print schema column name in front of each value**.

• Enregistrez le Job puis exécutez-le, en appuyant sur **F6**.
Le résultat obtenu en sortie en utilisant la méthode RPC GetStateName et correspondant au quarante-deuxième État des États-Unis est donc South Dakota.
tXSDValidator

Ce composant vérifie la qualité et la structure des données du fichier ou du flux à traiter.

Le tXSDValidator valide un fichier XML ou un flux XML entrant en fonction d'un fichier XSD et envoie les informations de validation à la sortie définie.

**Propriétés du tXSDValidator Standard**

Ces propriétés sont utilisées pour configurer le tXSDValidator s'exécutant dans le framework de Jobs Standard.

Le composant tXSDValidator Standard appartient à la famille XML.

Le composant de ce framework est toujours disponible.

**Basic settings**

| Mode                        | Sélectionnez le mode de validation dans la liste déroulante :
|                            | - *File Mode* pour valider un fichier d’entrée,
|                            | - *Flow Mode* pour valider un flux d’entrée.
|                            |
| Schema et Edit schema       | Un schéma est une description de lignes, il définit le nombre de champs qui sont traités et passés au composant suivant.
|                            | Notez que, lorsque le mode sélectionné est *File Mode*, le schéma de ce composant est en lecture seule et contient les informations standard de validation de fichier.
|                            |
| XSD file                   | Spécifiez le chemin d’accès au fichier XSD de référence.
|                            | Ce champ est disponible uniquement lorsque l’option *File Mode* est sélectionnée dans la liste *Mode*.
|                            |
| XML file                   | Spécifiez le chemin d’accès au fichier XML à valider.
|                            | Ce champ est disponible uniquement lorsque l’option *File Mode* est sélectionnée dans la liste *Mode*.
|                            |
| If XML is valid, display   | Saisissez le message à afficher dans la console si le fichier XML est valide.
|                            | Ce champ est disponible uniquement lorsque l’option *File Mode* est sélectionnée dans la liste *Mode*.
|                            |
| If XML is invalid, display | Saisissez le message à afficher dans la console si le fichier XML est invalide.
|                            | Ce champ est disponible uniquement lorsque l’option *File Mode* est sélectionnée dans la liste *Mode*.
|                            |
| Print to console           | Cochez cette case pour afficher le message de validation dans la console.
Ce champ est disponible uniquement lorsque l’option **File Mode** est sélectionnée dans la liste **Mode**.

### Allocate

Cliquez sur le bouton [+] pour ajouter autant de lignes que nécessaire et, dans chaque ligne, spécifiez la valeur des colonnes suivantes.

- **Input Column**: cliquez dans la cellule et sélectionnez une colonne à valider.
- **XSD File**: saisissez le chemin d'accès au fichier XSD de référence.

Ce champ est disponible uniquement lorsque l’option **File Mode** est sélectionnée dans la liste **Mode**.

### Advanced settings

#### Enable Features

Cliquez sur le bouton [+] pour ajouter autant de lignes que nécessaire, et, dans chaque ligne, saisissez la Feature à activer dans le parseur sous-jacent, entre guillemets doubles, par exemple, "http://apache.org/xml/features/honour-all-schemaLocations".


#### Encoding

Saisissez, entre guillemets doubles, le type d’encodage.

#### tStatCatcher Statistics

Cochez cette case pour collecter les métadonnées de traitement du Job, aussi bien au niveau du Job qu’au niveau de chaque composant.

### Global Variables

**Global Variables**

- **ERROR_MESSAGE**: message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable *After* et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case **Die on error** est décochée, si le composant a cette option.

- **DIFFERENCE**: résultat de la validation. Cette variable est une variable *Flow* et retourne une chaîne de caractères.

- **VALID**: résultat de la validation. Cette variable est une variable *Flow* et retourne un booléen.

- **XSD_ERROR_MESSAGE**: message d’erreur XSD généré par le composant. Cette variable est une variable *Flow* et retourne une chaîne de caractères.

Une variable *Flow* fonctionne durant l’exécution d’un composant. Une variable *After* fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.
Pour plus d'informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

**Utilisation**

**Règle d’utilisation**

Lorsque le mode sélectionné est **File Mode**, ce composant peut être utilisé en standalone mais il est généralement connecté à un composant de sortie pour rassembler les informations de validation.

---

**Scénario : Valider des flux de données par rapport à un fichier XSD**

Ce scénario décrit un Job qui valide une colonne XML dans le fichier d’entrée `ShipOrder.csv` par rapport au fichier XSD de référence `ShipOrder.xsd`, écrit les lignes valides dans le fichier délimité `ShipOrder_Valid.csv` et les lignes invalides ainsi que les messages d’erreur dans le fichier délimité `ShipOrder_Invalid.csv`.

Pour un exemple similaire d’utilisation validant un fichier XML, consultez Valider un fichier XML à la page 702.

Le contenu du fichier d’entrée `ShipOrder.csv` comprenant la colonne XML `ShipOrder` à valider se présente comme suit :

```
ID;ShipOrder
000001;<shiporder orderid="000001"><orderperson>George Bush</orderperson><shipto><name>John Adams</name><address>Oxford Street</address></shipto><item><title>Empire Burlesque</title><note>Special Edition</note><quantity>1</quantity><price>10.90</price></item></shiporder>
000002;<shiporder orderid="000002"><orderperson>Judy Liu</orderperson><shipto><name>Jack Liu</name><address>Wangfujing Street</address></shipto><item><title>Hide Your Heart</title><quantity>1</quantity><price>9.90</price></item></shiporder>
000003;<shiporder><orderperson>Peter Qian</orderperson><shipto><name>Thomas Wang</name><address>Wangfujing Street</address></shipto><item><title>The Power of Habit</title><quantity>1</quantity><price>8.99</price></item></shiporder>
```

Le contenu du fichier XSD de référence `ShipOrder.xsd` se présente comme suit :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="shiporder">
  
  </xs:element>
</xs:schema>
```
Construire le Job

Procédure

1. Créez un nouveau Job et ajoutez un composant tFileInputDelimited, un tXSDValidator et deux tFileOutputDelimited en saisissant leur nom dans l’espace de modélisation graphique ou en les déposant depuis la Palette.

2. Double-cliquez sur le composant tXSDValidator pour ouvrir sa vue Basic settings et sélectionnez Flow Mode dans la liste déroulante Mode.

3. Reliez le tFileInputDelimited au tXSDValidator à l’aide d’un lien Row > Main.

4. Reliez le tXSDValidator au premier tFileOutputDelimited à l’aide d’un lien Row > Main pour écrire en sortie les lignes valides.

5. Reliez le tXSDValidator au second tFileOutputDelimited à l’aide d’un lien Row > Rejects pour écrire en sortie les lignes invalides.

Configuration des composants

Procédure

1. Double-cliquez sur le tFileInputDelimited pour ouvrir sa vue Component.
2. Dans le champ **File name/Stream**, spécifiez le chemin d'accès au fichier d'entrée. Dans cet exemple, le fichier d'entrée est \E:/ShipOrder.csv. Dans le champ **Header**, saisissez 1 pour ignorer la première ligne (l'en-tête) du fichier d'entrée. Cliquez sur le bouton [...] à côté du champ **Edit schema** et définissez le schéma en ajoutant deux colonnes **ID** et **ShipOrder** de type **String**.

![Schema of tFileInputDelimited_1](image)

3. Double-cliquez sur le composant **tXSDValidator** pour ouvrir sa vue **Component**.

![tXSDValidator_1](image)

4. Cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant **tFileInputDelimited** précédent et, dans la boîte de dialogue qui s'ouvre, cliquez sur **Yes** pour propager le schéma aux deux composants **tFileOutputDelimited**.
Ajoutez une ligne à la table Allocate en cliquant sur le bouton [+]. Cliquez dans la cellule de la colonne Input Column et sélectionnez la colonne XML ShipOrder à valider. Dans la colonne XSD File, saisissez le chemin d'accès au fichier XSD de référence, E:/ShipOrder.xsd dans cet exemple.

5. Double-cliquez sur le premier tFileOutputDelimited pour ouvrir sa vue Component.

6. Dans le champ File Name, spécifiez le chemin d'accès au fichier de sortie qui stockera les lignes valides. Dans cet exemple, le fichier est E:/ShipOrder_Valid.csv. Cochez la case Include Header pour inclure les en-têtes de colonnes dans le fichier de sortie.

7. Double-cliquez sur le second tFileOutputDelimited pour ouvrir sa vue Component.

8. Cliquez sur le bouton [...] à côté du champ Edit schema pour voir son schéma.

Vous pouvez voir qu'une colonne supplémentaire errorMessage contenant les informations d'erreur pour les lignes invalides est ajoutée automatiquement au schéma, en plus des deux colonnes propagées.
9. Dans le champ **File Name**, spécifiez le chemin d'accès au fichier de sortie qui va stocker les lignes invalides et les messages d’erreur. Dans cet exemple, le fichier est `E:/ShipOrder_Invalid.csv`. Cochez la case **Include Header** pour inclure les colonnes d’en-tête dans le fichier de sortie.

**Sauvegarder et exécuter le Job**

**Procédure**

1. Appuyez sur les touches **Ctrl+S** afin de sauvegarder le Job.
2. Appuyez sur **F6** pour exécuter le Job.

Comme dans la capture d’écran ci-dessus, le fichier de sortie `ShipOrder_Valid.csv` contient deux lignes valides et le fichier de sortie `ShipOrder_Invalid.csv` contient une ligne invalide sans définition de l’attribut `orderid` et le message d’erreur.
tXSLT

Ce composant transforme une structure de données en une autre structure.
Le tXSLT tient compte d’une feuille de style XSL pour transformer un fichier source XML et l’envoyer vers la sortie définie.

**Propriétés du tXSLT Standard**

Ces propriétés sont utilisées pour configurer le tXSLT s’exécutant dans le framework de Jobs Standard.
Le composant tXSLT Standard appartient à la famille XML.
Le composant de ce framework est toujours disponible.

**Basic settings**

<table>
<thead>
<tr>
<th>Basic settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XML file</strong></td>
<td>Chemin d’accès au fichier XML à transformer.</td>
</tr>
<tr>
<td><strong>XSL file</strong></td>
<td>Chemin d’accès au fichier XSL de référence.</td>
</tr>
<tr>
<td><strong>Output file</strong></td>
<td>Chemin d’accès du fichier de sortie. Si le fichier n’existe pas, il sera créé. Le fichier de sortie peut être n’importe quel fichier structuré ou non tels que html, xml, txt ou même pdf ou edifact en fonction de votre fichier xsl.</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>Cliquez sur le bouton [+] pour ajouter de nouvelles lignes à la liste Parameters et définir les paramètres de transformation du fichier XSLT. Pour chaque ligne, renseignez la clé dans la colonne name et sa valeur associée dans la colonne value.</td>
</tr>
</tbody>
</table>

**Advanced settings**

<table>
<thead>
<tr>
<th>Advanced settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tStatCatcher Statistics</strong></td>
<td>Cochez cette case pour collecter les données de log au niveau du composant.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Global Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERROR_MESSAGE</strong></td>
<td>message d’erreur généré par le composant lorsqu’une erreur survient. Cette variable est une variable After et retourne une chaîne de caractères. Cette variable fonctionne uniquement si la case Die on error est décochée, si le composant a cette option.</td>
</tr>
<tr>
<td><strong>OUTPUT_FILEPATH</strong></td>
<td>chemin d’accès au fichier de sortie. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
<tr>
<td><strong>OUTPUT_FILENAME</strong></td>
<td>nom du fichier de sortie. Cette variable est une variable Flow et retourne une chaîne de caractères.</td>
</tr>
</tbody>
</table>
Une variable **Flow** fonctionne durant l’exécution d’un composant. Une variable **After** fonctionne après l’exécution d’un composant.

Pour renseigner un champ ou une expression à l’aide d’une variable, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables. A partir de cette liste, vous pouvez choisir la variable que vous souhaitez utiliser.

Pour plus d’informations concernant les variables, consultez le Guide utilisateur du Studio Talend.

### Utilisation

<table>
<thead>
<tr>
<th>Règle d’utilisation</th>
<th>Ce composant peut être utilisé en standalone.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitation</td>
<td>Du fait d’une incompatibilité de licence, un ou plusieurs Jar requis pour utiliser ce composant ne sont pas fournis. Vous pouvez installer les Jar manquants pour ce composant en cliquant sur le bouton <strong>Install</strong> dans l’onglet <strong>Component</strong>. Vous pouvez également trouver les Jar manquants et les ajouter dans l’onglet <strong>Modules</strong> de la perspective <strong>Integration</strong> de votre studio. Vous pouvez trouver plus d’informations concernant l’installation des modules externes dans Talend Help Center (<a href="https://help.talend.com">https://help.talend.com</a>).</td>
</tr>
</tbody>
</table>

### Scénario : Transformer un fichier XML en HTML à l’aide d’une feuille de style XSL

Ce scénario décrit un Job à deux composants appliquant une feuille de style XSL à un fichier XML et génère un fichier HTML en sortie. Il permet également de définir un paramètre de transformation pour la feuille de style XSL afin de changer la couleur de l’arrière-plan de l’en-tête dans le document HTML créé en sortie.

### Procédure

**Procédure**

1. A partir de la **Palette**, cliquez et déposez un composant **tXSLT** et un **tMsgBox** dans l’espace graphique.

![Diagramme des composants](image)

2. Double-cliquez sur le **tXSLT** pour ouvrir la vue **Basic settings** et paramétrer le composant.
3. Dans le champ **XML file**, renseignez le chemin d'accès ou parcourez vos dossiers jusqu’au fichier XML à transformer. Dans cet exemple, il s’agit d’une liste de titres MP3 et leurs informations associées (nom de l’artiste, maison de disque...).

4. Dans le champ **XSL file** de la vue **Basic settings**, renseignez le chemin d’accès ou parcourez vos dossiers jusqu’au fichier XSL approprié.

5. Dans le champ **Output file**, renseignez le chemin d’accès ou parcourez vos dossiers jusqu’au fichier HTML.

Dans la zone **Parameters** de la vue **Basic settings**, cliquez sur le bouton [+] pour ajouter une ligne où définir le nom et la valeur du paramètre de transformation du fichier XSL. Dans cet exemple, le nom du paramètre de transformation utilisé est *bgcolor* et sa valeur est *green*.

8. Double-cliquez sur le tMsgBox pour afficher la vue **Basic settings** et paramétrer le composant.

9. Enregistrez votre Job puis appuyez sur **F6** pour l'exécuter. Une boîte de dialogue s'affiche et vient confirmer que le fichier HTML est bien créé en sortie et qu'il est stocké à l'endroit préalablement défini.
10. Cliquez sur OK pour fermer la boîte de dialogue.

**Résultats**

Vous pouvez désormais ouvrir le fichier HTML créé pour vérifier la transformation des données XML et la couleur d’arrière-plan de l’en-tête du tableau.

**My CD Collection**

<table>
<thead>
<tr>
<th>Title</th>
<th>Artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empire Burlesque</td>
<td>Bob Dylan</td>
</tr>
<tr>
<td>Hide your heart</td>
<td>Bonnie Tyler</td>
</tr>
<tr>
<td>Greatest Hits</td>
<td>Dolly Parton</td>
</tr>
<tr>
<td>Still got the blues</td>
<td>Gary Moore</td>
</tr>
</tbody>
</table>